MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

10

Cr$ 8.200

INCLUI OS JOGOS, CORRIGIDOS, DAS EDIÇÕES 2 E 3.

A impressora Grafix 100MX

O poder da flor • Modems

JÁ ESTÁ À VENDA A CAPA DURA PARA ENCADERNAR O PRIMEIRO VOLUME DE MICROCOMPUTADOR CURSO PRÁTICO

rioGráfica

MANAUS, ALTAMIRA, MACAPÁ, JI.PARANÁ, PORTO VELHO, RIO BRANCO, SANTARÉM, BOA VISTA, SINOP, ALTA FLORESTA: Cr$ 10.660

MICROCOMPUTADOR CURSO PRÁTICO

ATENÇÃO

Nesta edição, você encontra, reimpressas, as páginas de 9 a 16 de Jogos, que saíram com listagens incorretas. Favor substituí-las na sua coleção, eliminando as anteriores, que saíram nas edições 2 e 3 de MICROCOMPUTADOR — CURSO PRÁTICO.

APLICAÇÕES

Dados a distância 197

PERIFÉRICOS

O poder da flor 200

SOFTWARE

1-2-3: macrocomandos 202

PROJETOS DE PROGRAMAS

Psiquiatra eletrônico 204

RAIO X

Grafix 100MX 206

CIÊNCIA DA COMPUTAÇÃO

A resposta lógica 208

LINGUAGENS

Dados e decisões 211

PERFIL

Elegância italiana 216

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1) Qual a principal vantagem da impressora tipo margarida em relação à matricial? E a principal desvantagem?
2) Quais são as quatro leis básicas da álgebra booleana?
3) Qual a diferença entre um modem de conexão direta e um acoplador acústico?
4) Qual a instrução condicional do PASCAL que lida com opções múltiplas?

RESPOSTAS DO TESTE ANTERIOR

1) A linguagem que teve mais influência na concepção do PASCAL foi o ALGOL 60.
2) A expressão inglesa que qualifica um micro (ou um programa) que tem boa interação com o usuário é "user-friendly" (literalmente, "amigável com o usuário").
3) O recurso disponível nos softwares integrados Lotus 1-2-3 e Symphony para armazenar seqüências de comandos freqüentemente utilizadas são os macrocomandos.
4) O símbolo Ω significa ohm, que é a unidade de medida da resistência.
5) Os símbolos visuais que representam as diversas opções num menu de tela são chamados "ícones".

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 terá doze páginas e quatro capas, e incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as 3.ª e 4.ª trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 18 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem as 3.ª e 4.ª capas do fascículo 30); 3.ª e 4.ª capas dos fascículos de 1 a 29; e guardas.

rioGráfica **Diretoria Executiva:** Oscar Neves; Filipe Zander; Nilo Sérgio de Almeida; Danilo Esteves Costa; **GRUPO EDUCAÇÃO E CULTURA - Planejamento e Comercialização - Gerente de Planejamento:** João Azeredo Coutinho; **Gerente de Marketing:** Jaime Rodrigues; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Aylton Menezes; **Diretor de Vendas:** Marcos March; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto: **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Maurício Rittner; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Revisores:** Isis Augusta Loyolla; Cacilda Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Aníbal Monteiro; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistente de Arte:** Miguel Luis Escámez Simón; **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; Claudio Marcondes; Nadya Morelo Reis; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Fernando Dorfman Knijnik; Lúcia Maria Leal Gonçalves; Maria Clara Cescato; Maria Ester Menezes Martinho; Newton Roberval Eichemberg; Patrícia de Paiva e Castro; Regina Amarante; **Pesquisa:** Stella Maria Quentel; **Arte:** Toshio Morobosi; **Assessoria Técnica:** Geraldo Coen; Gustavo José Ferreira Grubba; Lauro de Lauro; Pierluigi Piazzi; Sérgio Jorge Batarce; Imarés Microcomputadores. As fotos não creditadas pertencem à obra original. Seguem-se os créditos das demais: Olivetti; Fernando Scavone; Jorge's Estúdio.

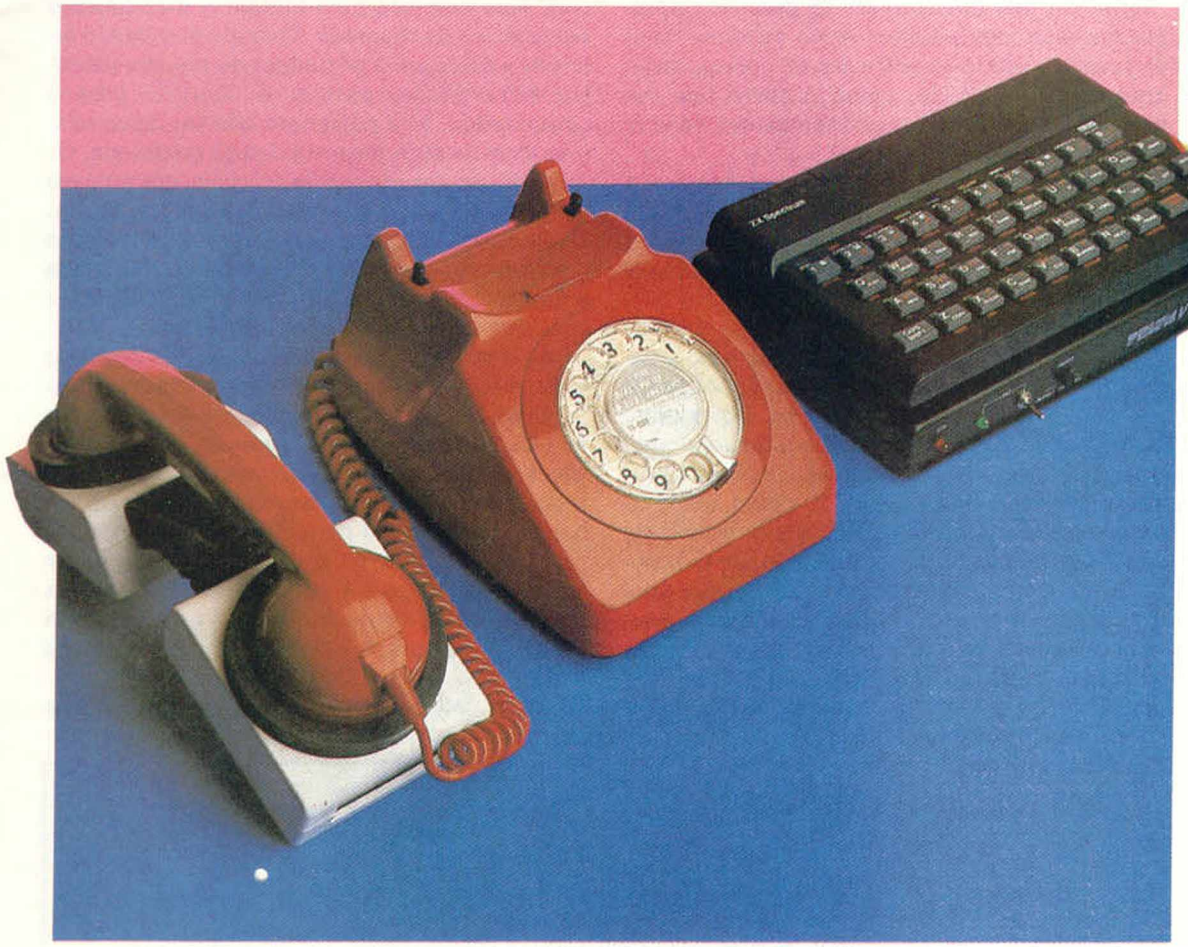
© APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa.

Composição: Editora Rio Gráfica Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG.

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Ltda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

DADOS A DISTÂNCIA



O tom de comando
Desenvolvido originalmente para conectar terminais distantes a computadores de grande porte, o modem converte (ou modula) dados digitais em tons acústicos para transmissão telefônica e, na recepção, demodula os tons acústicos de volta em sinais digitais. Os acopladores acústicos transmitem e recebem por meio do aparelho telefônico, enquanto os modems de conexão direta se ligam à linha telefônica.

"Telecomunicações" é o termo usado para descrever o processo de enviar informações (tanto de voz como de dados) pela rede telefônica. Nesse sentido, o modem revela-se um dispositivo de grande utilidade.

A conexão de seu micro a um modem lhe possibilita comunicar-se com outros computadores pela linha telefônica. Assim, muitas informações são acessadas. Você pode mandar cartas, que serão recebidas no instante em que forem transmitidas, intercambiar software com amigos distantes e mensagens com muitos outros usuários de microcomputadores, além de acessar computadores de grande porte.

A tecnologia de telecomunicação tem sua origem nos computadores de grande porte. Numa configuração típica, a CPU de um grande computador está numa sala especialmente construída, com ar condicionado, e se conecta a terminais espalhados por todo o prédio. Um terminal compõe-se apenas de uma tela e um teclado ligados ao computador central por meio de um cabo serial. Dessa forma, o usuário pode utilizar um terminal num extremo do prédio e acessar o computador no outro extremo.

Uma ligação por cabo entre o computador e o terminal funciona bem para distâncias relativamente curtas — no máximo, algumas centenas de metros —, mas a deterioração do sinal exclui o uso desse sistema para distâncias maiores. Para resolver esse problema, desenvolveu-se o modem (*mo*dulador/*demo*dulador).

Esse dispositivo permite que dados sejam transmitidos por uma linha telefônica comum. Ele transforma os sinais digitais em tons acústicos (sinais analógicos) de freqüência e volume apropriados para a transmissão pela rede telefônica. O processo é conhecido por "modulação". O modem do computador que recebe a transmissão reconverte esses tons acústicos de volta em sinais digitais que passam para o computador receptor. É a demodulação. Usa-se um

sinal constante (chamado "sinal de portadora") como referência, enquanto o dado é transmitido na onda senóide (modulada).

Como resultado desse processo, acessam-se computadores distantes quase como se estivessem diretamente conectados ao terminal. Os terminais dedicados raramente têm poder próprio de processamento, sendo então chamados de "terminais não-inteligentes". É possível empregar um microcomputador como terminal, simplesmente usando-se software de comunicação apropriado. Contudo, como o micro tem sua própria capacidade de processamento, é chamado de "terminal inteligente".

Existem dois tipos de modems: o acoplador acústico e o modem de conexão direta. O acoplador acústico é mais simples: tem dois bocais de borracha em que se coloca o gancho do telefone. Os sinais acústicos são transmitidos pelo bocal e recebidos pelo fone. O modem acústico é, dos dois tipos, o mais conveniente para sistemas portáteis e microcomputadores, pois pode ser usado com qualquer telefone. Há mesmo quem use modems acionados a pilha com computadores portáteis, fazendo-se a ligação a partir de telefones públicos. Por outro lado, os modems acústicos estão sujeitos a interferência (ruídos do ambiente) e, por isso, não são tão confiáveis. Os modems de conexão direta têm quatro saídas: duas para a linha telefônica e duas para os fios do telefone propriamente dito. Além de serem mais confiáveis que seus equivalentes acústicos, os modems de conexão direta em geral oferecem mais recursos.

No Brasil, os modems são aprovados pela Embratel e sua fabricação segue as normas internacionais de comunicação. Essas duas condições garantem que os modems funcionem de acordo com as normas telefônicas, evitando ruídos ou perdas de sinais.

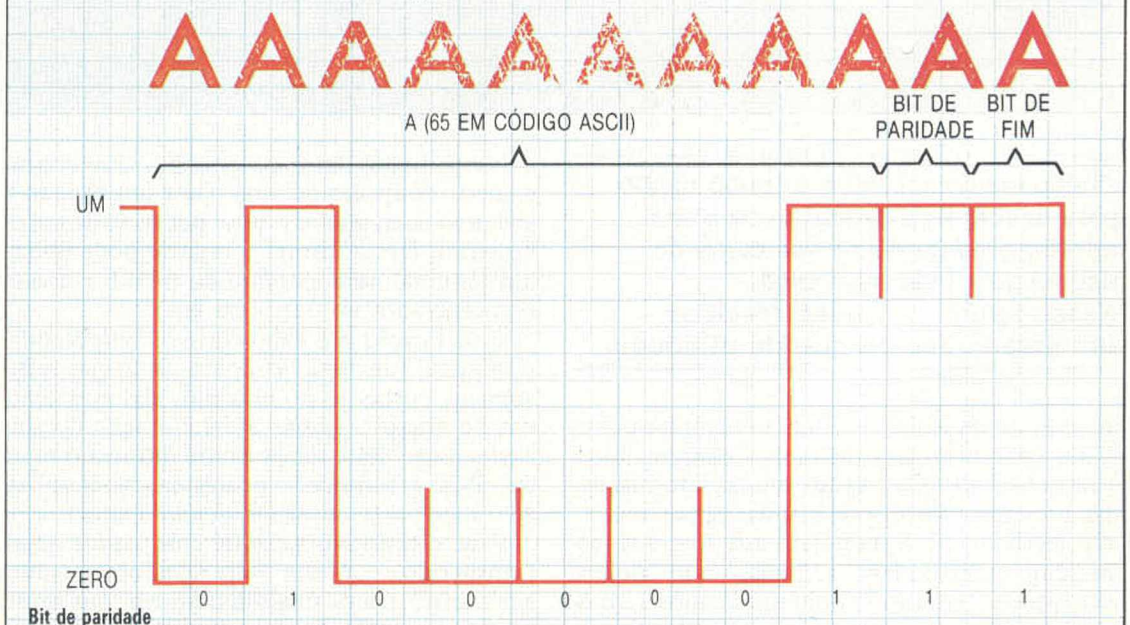
Alguns modems oferecem recursos como "auto-resposta" e "autodiscagem". Os modems com auto-resposta possibilitam o atendimento automático da chamada. Quando atendem o telefone e detectam a portadora do modem da outra extremidade, passam o controle para o computador. Se a portadora não for detectada, o modem de auto-resposta desliga o telefone. Os modems de autodiscagem recebem um número do computador e automaticamente o discam. Dessa forma, em resposta a um nome digitado pelo usuário, o software do computador consulta o número do telefone num banco de dados e faz com que o modem disque tal número.

Mede-se a velocidade de transmissão de dados entre dois dispositivos em bauds. Normalmente, considera-se tal velocidade como o número de bits transmitidos por segundo, embora esta não seja uma definição precisa.

As velocidades de transmissão ou taxas de bauds mais comuns para dispositivos de comunicação são 300, 1.200 e 1.200/75 (esta se refere a dados recebidos a 1.200 bauds e transmitidos a 75). A maioria dos micros grava programas em cassete com uma velocidade dentro da mesma faixa (entre 300 e 1.200 bauds).

Vários padrões de comunicação são utilizados em computadores de pequeno e de grande por-

Transmissão e verificação



Bit de paridade

Transmitem-se os dados pela linha telefônica, em código binário correspondente a cada caractere, como uma seqüência de tons acústicos. A freqüência do 0 binário fica um pouco abaixo do tom de referência, ou sinal de portadora, ao passo que o 1 binário tem freqüência um pouco acima. Para se detectar erro, geram-se os bits de paridade.
Por exemplo: aqui, o caractere transmitido é o A—65 em código ASCII; 01000001 em binário. Há um número par de dígitos 1 neste código e a paridade é ímpar, o que significa que a soma dos dígitos 1 do caractere deve ter um número ímpar de uns; o bit de paridade é, assim, estabelecido como 1. Em seguida, há o bit de fim (stop), sinalizando o término da palavra transmitida. Essa palavra tem 8 bits, mas exige 10 bits para sua transmissão.

te. Podem variar a velocidade, o tamanho do bloco de informação transmitido, os bits de paridade e até mesmo o sistema telefônico.

65 = 01000001

Por causa das diferenças entre os sistemas telefônicos do Brasil e dos Estados Unidos, por exemplo, os dois países usam padrões de sinais telefônicos diferentes. O padrão brasileiro é conhecido como CCITT (ou V21), enquanto o equivalente americano é o Bell (nome derivado da companhia telefônica Bell).

Como o sistema telefônico só pode diferenciar um número limitado de freqüências com algum grau de confiabilidade, os dados são transmitidos em forma binária. Para tanto, cada caractere no código ASCII é traduzido para sua forma binária. A letra A converte-se do código 65 (ASCII) para o 01000001 (binário). Os dígitos 1 são representados por uma freqüência "alta", pouco acima do sinal de portadora; uma freqüência "baixa", pouco abaixo do sinal de portadora, representa os dígitos 0.

O computador que recebe a informação precisa identificar onde termina um caractere e começa o próximo. Usam-se, portanto, bits de início (start) e de fim (stop). Esses dois bits são delimitadores da palavra (conjunto de códigos ASCII) transmitida e há convenções que os estabelecem. Quando recebe um bit de fim, por exemplo, o computador decodifica os bits precedentes. O software que controla toda a comunicação é o protocolo. Sua função é disciplinar o meio de comunicação. Ele decodifica a palavra após receber o bit de fim (stop) e codifica os dados recebidos e transmitidos para a transmissão e verificação da paridade. O protocolo mais comum para comunicação em ASCII é o XON/XOFF, de 8 bits de informação seguidos por 1 bit de fim. Um esquema alternativo é o de 7 bits de dados e 2 bits de fim. Outros sistemas usam tanto 1 bit de início quanto 1 de fim — a mudança indica o fim do caractere.

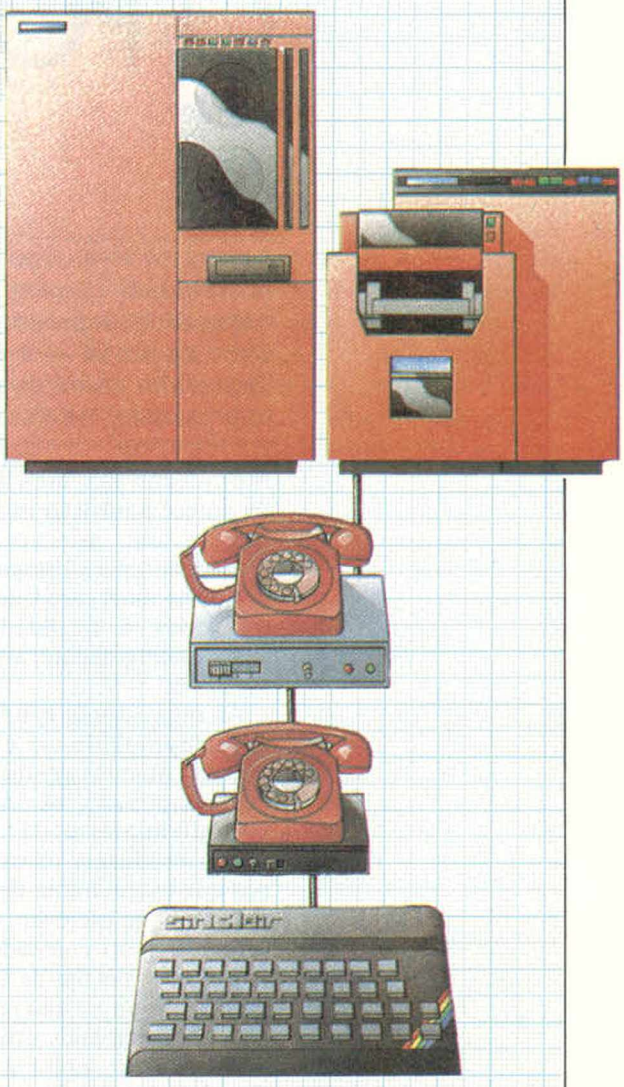
Uma vez que nenhuma comunicação é totalmente confiável, precisamos de uma forma para verificar se houve erro. A solução mais simples, conhecida como "verificação da paridade", consiste em contar o número de bits altos (1) e adicionar 1 bit extra de paridade para fazer com que o total de bits altos seja ímpar (num sistema de paridade ímpar) ou par (num sistema de paridade par). Por exemplo, voltando à letra A, o número total de bits altos em 01000001 é 2 (um número par). Dessa forma, para tornar a paridade ímpar, o bit de paridade também terá de ser alto. Já no caso da letra C — traduzida como 01000011 em código binário —, o bit de paridade terá de ser baixo.

A verificação de paridade é pouco sofisticada: detecta um erro num único bit, mas não detecta dois erros no mesmo caractere, estando a paridade certa. É, contudo, simples e útil para as aplicações em que a precisão não é vital.

Fluxo de informação

A taxa de bauds, assim denominada em homenagem a Émile Baudot, pioneiro francês da transmissão de dados, é uma medida da velocidade do fluxo de informações entre dispositivos de comunicação. A taxa de baud é 1 se a informação for transmitida a 1 bit por segundo. Tipicamente, dispositivos que se comunicam por modem usam taxas que variam entre 300 e 9.600 bauds.

No estabelecimento da taxa de bauds, devem-se considerar os bits de início, de fim e de paridade. Por exemplo: num sistema em que há verificação de paridade e que usa 8 bits de dados e 1 bit de fim para cada caractere, apenas 8 em cada 10 bits transmitidos contêm informações úteis. Assim, numa velocidade de transmissão de 300 bauds, o verdadeiro número de bits por segundo é 240 (80% de 300).



Registre

Linhas Apple e TRS-80

TÚNEL COLORIDO

Este programa em BASIC produz belo efeito visual na tela: um túnel que se afunila e muda de forma, produzindo impressão tridimensional. As cores mudam aleatoriamente, intensificando o efeito.

```
5  REM *** TUNEL COLORIDO ***
10 GR : HOME
20 FOR I =1 TO 5
30 FOR N = 0 TO 39
40 W = INT ( RND (1) * 15 )
50 COLOR = W
60 VLIN 0,39 AT N
70 VLIN 0,39 AT 39 - N
80 HLIN 0,39 AT N
90 HLIN 0,39 AT 39 - N
100 NEXT N, I
110 END
```

SINAIS

Digital:
Forma de onda quadrada, na qual existem apenas dois níveis: 0 V e +5 V. É o sinal utilizado internamente nos computadores.

Analógico:
Pode ser descrito por uma senóide, em que é aceita qualquer variação, em freqüência, amplitude ou fase.

O PODER DA FLOR

As impressoras tipo margarida apresentam funções práticas — como espaçamento proporcional entre as letras — e qualidade de impressão muito superior à das impressoras matriciais.

À primeira vista, uma impressora tipo margarida parece aquisição estranha para o proprietário de um microcomputador. Além de inadequada para listagens de programas, é lenta e custa mais caro que uma impressora matricial. Mas a qualidade de impressão desse equipamento encontra-se muito à frente de seus similares.

As impressoras matriciais formam os caracteres imprimindo um conjunto de pontos: não importa quantas agulhas sejam usadas na cabeça de impressão, os pontos sempre aparecerão no texto impresso. Já as impressoras tipo margarida produzem os caracteres por meio de tipos que batem numa fita entintada — como nas máquinas de escrever. Os tipos, um para cada caractere, são colocados em círculo, assemelhando-se a pétalas de flor — daí o nome "margarida". A impressão resultante é de leitura mais fácil e parece mais profissional.

No entanto, a alta qualidade tem sua contrapartida: a impressão apresenta a grande desvantagem da pequena velocidade. Para imprimir um caractere, a margarida gira até a "pétala" desejada ficar no alto; daí, o martelo de impressão bate no tipo e o carro se move para produzir outro caractere.

Comparando esse processo com o da impressora matricial, que imprime os pontos à medida que o carro anda de um lado para outro do papel, entende-se a razão da diferença de velocidade. Uma impressora tipo margarida padrão imprime cerca de vinte caracteres por segundo (cps) — há mais velozes, de oitenta cps, mas custam muito mais caro; uma impressora matricial Epson FX-86, por exemplo, custa o mesmo, mas tem velocidade de 160 cps.

Duas funções extras permitem aumentar a velocidade das impressoras tanto matriciais como do tipo margarida: impressão bidirecional e procura lógica. Bidirecional significa apenas que uma linha do texto é impressa da esquerda para a direita enquanto a linha seguinte o é da direita para a esquerda. A impressora não precisa esperar o carro voltar e, assim, pode imprimir com maior rapidez. Procura lógica significa que o carro pula espaços para chegar à palavra seguinte do texto — impressoras menos sofisticadas levam o mesmo tempo para imprimir um espaço ou qualquer caractere.

As formas dos caracteres de uma impressora matricial estão armazenadas na ROM da impressora; os caracteres das impressoras tipo margarida estão armazenados na própria margarida, em forma de tipos físicos. Cada método tem suas vantagens: com a matriz de pontos, as formas dos caracteres podem ser redefinidas enviando-se códigos adequados para a impressora a partir do micro. Na impressora tipo margarida, esse processo é muito mais fácil: basta trocar a margarida por outra, diferente.

As margaridas possuem grande variedade de modelos de tipos e pitches; o modelo de tipo refere-se ao desenho dos caracteres, enquanto o pitch diz respeito à largura do caractere. Os modelos mais comuns de tipo são: courier, romano, gótico e itálico. O pitch, em geral, é de dez ou doze caracteres por polegada.

O único problema com os diversos modelos é que poucos têm os caracteres exatamente iguais aos usados por um micro. Por exemplo, o caractere especial # sai, em geral, impresso com o símbolo do cifrão ($), um colchete ([) aparece no papel como a fração 1/2. Em muitos modelos, não se pode distinguir o número 0 (zero) da letra O maiúscula. Da mesma forma, a letra l minúscula pode ser confundida com o número 1. Enquanto as impressoras mais caras possuem margaridas de 127 caracteres, a maioria imprime apenas 92 ou 96 caracteres, e são estas as que costumam apresentar esse tipo de problema.

Imagine como se torna muito mais difícil depurar um programa quando é impossível distinguir o número 1 da letra l. Por isso, e também por sua lentidão, não se recomenda a aquisição de uma impressora tipo margarida para quem usa o micro principalmente em programação.

Efeitos especiais

À semelhança das matriciais, as impressoras tipo margarida podem ser programadas para produzir efeitos especiais, usando os mesmos métodos. O número desses efeitos, no entanto, é limitado. Por exemplo, alguns modelos dispõem de códigos especiais para ativar e desativar o sublinhado automático, marcar um ponto tabulado, determinar a margem esquerda e empurrar o papel meia linha para cima ou para baixo — permitindo a impressão de caracteres subscritos ou superescritos.

As impressoras tipo margarida também contam com recursos que possibilitam efeitos equi-

valentes ao texto em negrito de uma impressora matricial, em que cada caractere é reimpresso quatro vezes, para ficar destacado. Esse tipo de escrita, porém, apresenta a desvantagem de reduzir bastante a velocidade de impressão.

Em alguns modelos é possível variar a distância entre os caracteres e o espaço entre as linhas. Nesse caso, a impressora serve para criar imagens gráficas e dumps de tela. Se um pixel de tela está "ligado", a margarida imprime um ponto; se estiver "desligado", imprime um espaço. Reduzindo a distância entre os caracteres, pode-se passar uma linha horizontal inteira da tela para o papel. Da mesma forma, reduz-se o intervalo das linhas para não deixar espaço entre elas. Contudo, esse processo é muito lento.

Uma característica não encontrada nas impressoras matriciais é a capacidade de centralização automática dos títulos. Outro aspecto favorável é a tabulação decimal, que alinha as vírgulas decimais de uma lista de números, facilitando a adição de valores monetários.

Espaçamento estético

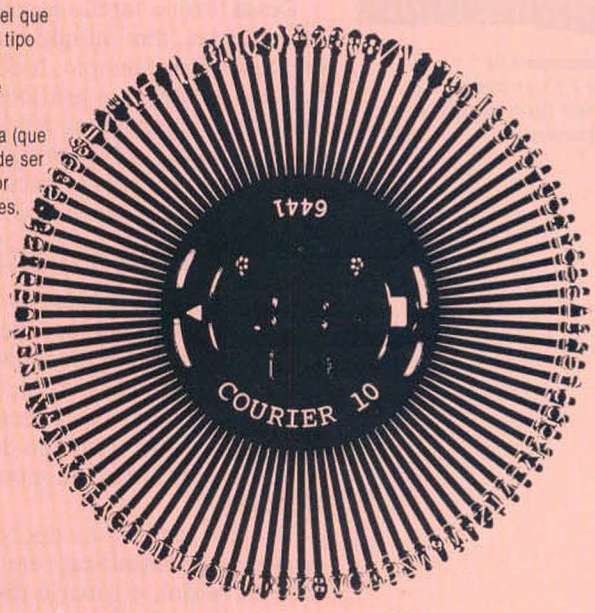
As impressoras tipo margarida mais caras realizam espaçamento proporcional. O pitch não é mais padronizado, com dez ou doze caracteres por polegada, mas varia conforme a largura do caractere impresso. Por exemplo, o caractere w é muito mais largo que o i; com espaçamento proporcional, o carro não avança tanto para um i quanto avançaria para um w. Isso significa que os caracteres em linhas sucessivas do texto não ficam exatamente uns embaixo dos outros, e o aspecto geral é mais agradável aos olhos — como as linhas desta página. Na maioria das vezes, precisa-se de uma margarida especial para usar corretamente essa função, mas ela é ligada e desligada por meio de códigos — como em qualquer outro efeito da impressora.

Num computador comercial, o uso desse tipo de impressora justifica-se plenamente, o que não ocorre no caso dos usuários de micros domésticos. Uma alternativa é a adaptação de uma máquina de escrever eletrônica, menos cara que a impressora tipo margarida, embora use o mesmo método de impressão.

Até há pouco tempo, as máquinas de escrever eletrônicas não podiam ser ligadas a um micro, pois não dispunham de interface ou saída serial RS232. Contudo, alguns modelos lançados na metade da década de 80 já saem da fábrica com interface embutida, ou permitem adaptar um kit de interface. Além do preço mais acessível, a vantagem de se utilizar essas máquinas eletrônicas é que elas continuam funcionando como máquinas comuns. Assim, você terá uma máquina para escrever cartas personalizadas ou endereçar envelopes e, ao mesmo tempo, uma impressora para o processamento de textos. No Brasil, dada a inexistência de modelos comerciais de impressoras tipo margarida, a única alternativa à importação é adaptar uma das máquinas eletrônicas existentes como, por exemplo, a Olivetti ET 121, a Olympia Dismac OAT 1200 etc.

Recursos da margarida

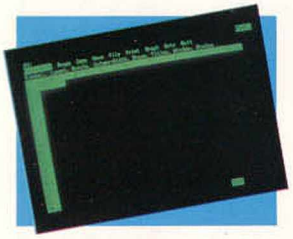
```
AMOSTRA DE IMPRESSÃO COM 10 CARACTERES POR POLEGADA
Amostra de impressão com 10 caracteres por polegada
AMOSTRA DE IMPRESSÃO COM 12 CARACTERES POR POLEGADA
Amostra de impressão com 12 caracteres por polegada
AMOSTRA DE IMPRESSÃO COM 15 CARACTERES POR POLEGADA
Amostra de impressão com 15 caracteres por polegada

O USO DE NEGRITO DESTACA PARTES DO TEXTO
O uso de negrito destaca partes do texto

                    TEXTO CENTRALIZADO
                    Texto Centralizado

Com espaçamento proporcional, obtém-se por exemplo, w com a
mesma largura de i :

wwwwwwwwwwwwwwww
iiiiiiiiiiiiiiii

Trocando-se as margaridas obtemos diferentes tipos de caracteres

                    0123456789
                    0123456789
                    0123456789
```



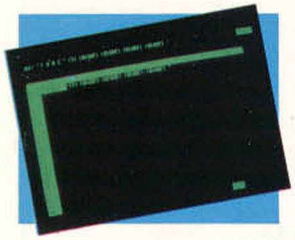
Mais cara e menos flexível que a matricial, a impressora tipo margarida imprime com qualidade de máquina de escrever e espaçamento proporcional. A margarida (que se parece com a flor) pode ser facilmente substituída por outra, com tipos diferentes.

1-2-3: MACROCOMANDOS

Ampliando os recursos do Lotus 1-2-3, os macrocomandos permitem programar esse software integrado de modo que ele adquire características de uma verdadeira linguagem.



A tela do Lotus
No Lotus, para trazer o menu principal à tela, basta teclar [/], como no Visicalc.



Macroeconomia
Este é o macrocomando para nomear faixas tal como aparece na planilha.

As planilhas projetadas para micros de uso pessoal sofrem restrições devido à pequena quantidade de RAM disponível para o programa e para os modelos financeiros desenvolvidos. Programas similares, escritos para o IBM PC, o Apricot da ACT e outros equipamentos profissionais, podem dispor de memória residente extensa e maior velocidade de processamento.

Em conseqüência, alguns dos softwares mais inovadores — a exemplo do Lotus 1-2-3, que integra planilha, banco de dados e gerador de gráficos — só rodam em máquinas caras, daqueles tipos.

No entanto, o 1-2-3 oferece ao usuário recursos úteis, como a possibilidade de programar com os macrocomandos.

O macro é uma seqüência de comandos freqüentemente utilizada. A representação gráfica desses comandos é armazenada na planilha, e recebe um nome especial, formado por uma barra invertida (\) e uma letra. Para executar a seqüência toda, basta pressionar [ALT] (no teclado do IBM PC e compatíveis) e mais a letra escolhida para o macro. Assim programado, o Lotus 1-2-3 roda sozinho, podendo executar um número teoricamente infinito de comandos.

Dessa forma, tarefas repetitivas podem ser automatizadas, diminuindo o tempo e os erros de digitação. Por exemplo, todos os cálculos de um balanço mensal, e os gráficos e relatórios que ele produz, são planejados detalhadamente, comando a comando, sob a forma de um ou mais macros. Feito isso, para executar a rotina mensal completa, basta pressionar algumas teclas.

O programa em ação

As linhas são designadas por letras e as colunas, numeradas. Dirigido por menus, o 1-2-3 mostra na tela os comandos disponíveis, quando se pressiona a tecla [/]. A partir desse ponto, as opções são selecionadas com a tecla correspondente à primeira letra do comando desejado, ou posicionando o cursor nesse comando e pressionando [Return].

O 1-2-3 tem tantas opções de comando que seus menus se ramificam em vários níveis de submenus. Assim, o usuário dispõe do Lotus 1-2-3 para realizar, literalmente, centenas de tarefas; mas isso também significa que algumas operações exigem muitos toques no teclado, como mostra o diagrama.

O 1-2-3 permite dar nome a uma célula ou faixa de células contíguas. Para operar com a faixa nomeada — incluí-la numa fórmula, por exemplo —, o nome é usado no lugar do endereço da célula, como segue:

A3 – B3 = C3	VENDAS – CUSTO = LUCRO
(endereços de células)	(nomes de células)

Dar nomes às regiões simplifica e acelera o processo de trabalhar com uma planilha extensa. Contudo, nomear um grupo de quatro células no 1-2-3 requer os seguintes comandos:

/ — mostra o menu principal.
R(ange) — diz ao programa que você vai realizar uma operação em número restrito de células e não na planilha inteira.
N(ame) — indica que você vai dar um nome à faixa especificada de células.
C(reate) — prepara o 1-2-3 para aceitar um nome e adotá-lo.
Digite o nome a ser adotado — CUSTO, por exemplo. Tecle [Return] para aceitar a célula indicada pelo cursor como começo da faixa. Mova o cursor quatro espaços para a direita. Tecle [Return] para aceitar a célula indicada pelo cursor como o fim da faixa.

Esse processo exige dez toques no teclado, sem contar a digitação do nome da faixa. Para reduzir o número de toques, podemos montar um macrocomando para nomear faixas automaticamente. Começa-se por fixar a localização dos macros na planilha, escolhendo uma seção de células para isso.

Tal escolha exige cuidados especiais, por dois motivos. Em primeiro lugar, as células devem ocupar uma região segura da planilha, uma área em que nunca serão colocados dados. Em segundo, o 1-2-3 só aloca na memória as células indicadas pelo cursor; assim, os espaços vazios entre as células de dados ocuparão lugar na memória.

Se o macro for colocado muito à direita do resto da planilha, inúmeros Kbytes de memória utilizável poderão ser consumidos por células vazias. Por essa razão, muitas vezes é preferível colocar os macros na extremidade esquerda da planilha. Considere um macro construído na coluna A. Se o número de comandos for excessivo e não couber numa célula, o macro poderá ficar contido em linhas sucessivas da mesma coluna. Posicionando o cursor na célula A1, digitam-se as iniciais dos comandos necessários, precedidas de um apóstrofo:

'/RNC

Nesse ponto, o 1-2-3 espera pela entrada do nome escolhido para a faixa. Digitando um ponto de interrogação entre chaves, introduz-se uma pausa. O programa aguarda até que o usuário pressione [Return], para então prosseguir com o macro. Utilizam-se palavras entre chaves, quando se trata de uma ação que não pode ser indicada por tecla específica, como os movimentos do cursor (designados por uma palavra de direção entre chaves). A indicação dos returns se faz por um til. O macro deve agora comandar o cursor para mover-se quatro células para a direita (right), cobrindo a faixa designada como "CUSTO". Seu formato ficará assim:

'/RNC{?}~{right}{right}{right}{right}~

Depois de dar entrada a todos os comandos necessários para nomear uma região, o corpo do macro está pronto.

O próximo passo consiste em nomear a região onde está armazenado o próprio macro, o que será realizado com uma tecla — por exemplo, [N], de nomear. Será necessário digitar novamente todos os comandos, pois o macro ainda não foi nomeado nem armazenado. Coloca-se, então, o cursor na célula A1 e digita-se:

[/][R][N][C][\][N][Return][Return]

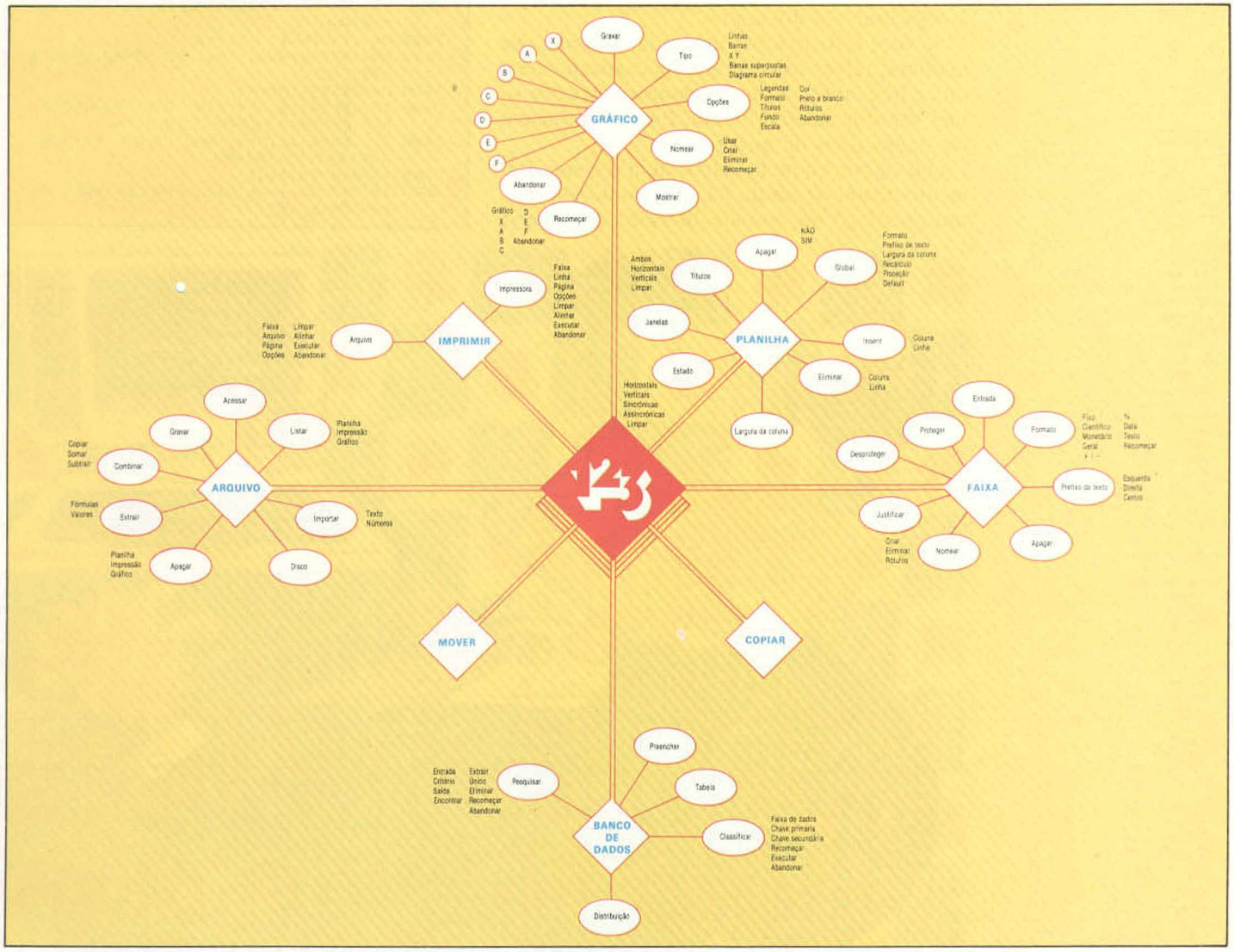
O caractere \ é usado para indicar que a tecla [ALT] deve ser pressionada; assim [\][N], o nome do macro, significa, para o 1-2-3, [ALT]+[N]. Agora que a região foi nomeada, todas as vezes que se pressionarem [ALT] e [N] simultaneamente, a seqüência de comandos armazenada será ativada. Desse momento em diante, para nomear uma faixa de quatro células, basta digitar:

[ALT][N] (nome da faixa) [Return]

Ou seja, há um grande avanço em relação ao procedimento inicial.

Esse exemplo, embora útil, é apenas uma pequena amostra do potencial dos macrocomandos.

O princípio dos macrocomandos permite que o usuário utilize aplicativos quase da mesma forma como alguém escreveria um programa BASIC ou PASCAL. Embora o Lotus 1-2-3 se limite, por enquanto, aos equipamentos de 16 bits, o conceito dos macros logo estará disponível em softwares destinados a equipamentos de uso pessoal.



A árvore de comandos Lotus
O Lotus tem vários níveis de menus. Estes comandos estão incorporados ao programa, enquanto outros podem ser definidos pelo usuário por meio dos macros.

PSIQUIATRA ELETRÔNICO

Simulando uma sessão de psicoterapia, o programa aqui analisado ilustra técnicas de manipulação de strings e dá uma idéia do grande desafio que representa a análise computacional da linguagem humana.

Um dos pioneiros na computação, Alan Turing, propôs um bom teste de inteligência artificial: se a máquina puder conversar com o ser humano sem que ele perceba que está dialogando com uma máquina, esta será "inteligente".

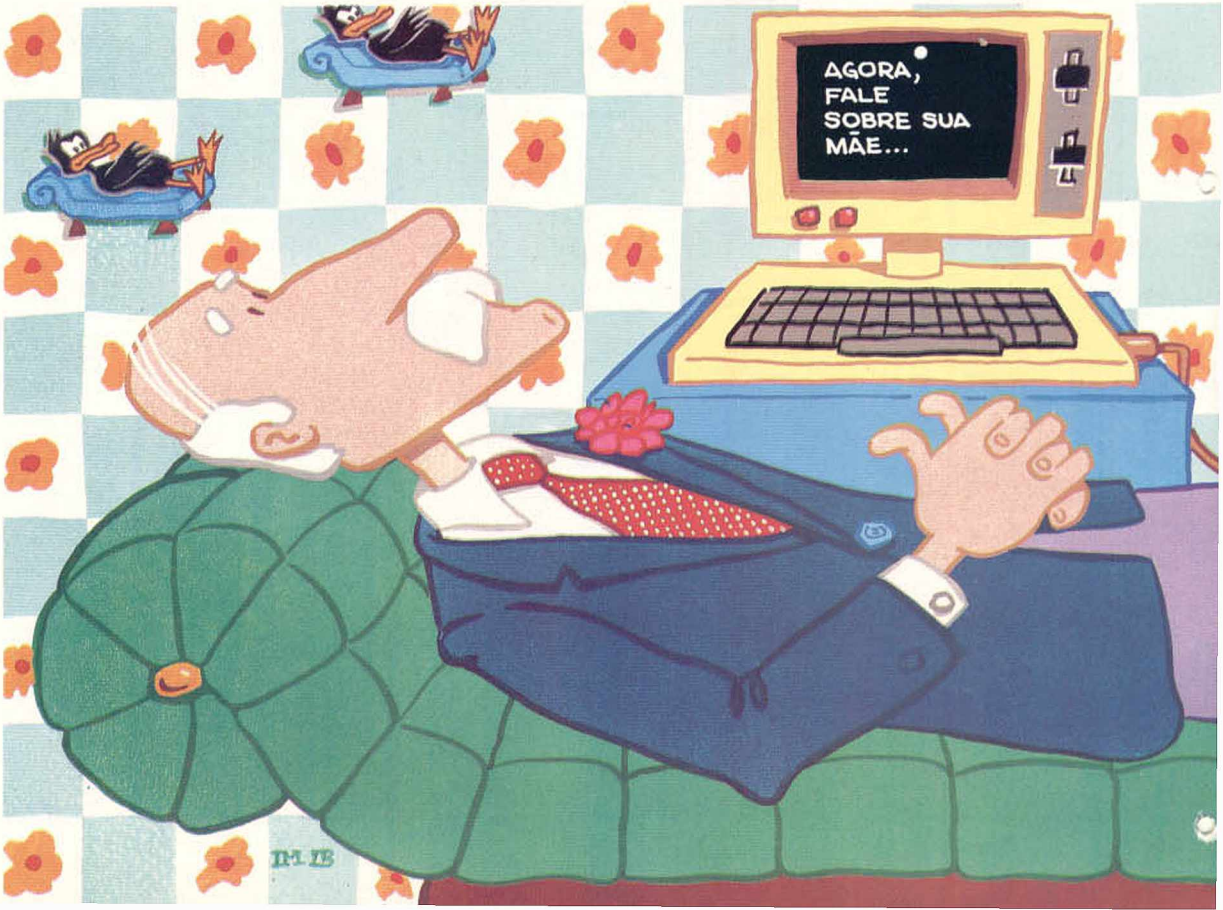
Até agora, nenhuma máquina passou nesse teste, embora vários programas aproximem-se da inteligência artificial. Não chegam a trocar opiniões com a pessoa, mas a estimulam a falar e a organizar seu discurso.

Se você refletir sobre uma boa conversa que manteve, talvez conclua que foi praticamente o único que falou. A outra pessoa apenas deu sinais sonoros de estar atenta. Esse é o princípio de alguns tipos de psicoterapia.

O programa aqui listado simula o comportamento de um terapeuta não direcional, ou seja, um psicólogo que jamais direciona a conversa

Análise

```
100  REM ********************************
110  REM *        PARA LINHA APPLE        *
110  REM *             ANALISE            *
120  REM ********************************
140  GOSUB 2000
200  FOR L=1 TO 2 * LT STEP 2
300  PRINT O$:INPUT I$
400  GOSUB 3000
500  NEXT L
1000 PRINT TAB(5);"---A SESSAO TERMINOU-
     --"
1100 PRINT"--DIGITE QUALQUER TECLA PARA
     CONTINUAR--"
1200 GET A$:IF A$="" THEN 1200
1300 GOSUB 5000
1900 END
1990 REM ********************************
2000 REM **          INICIALIZA          **
2010 REM ********************************
2050 LT=20:AN=10:T9=500:EX=2*LT
2100 DIM R$(AN):DIM H$(2*LT)
2200 DATA "SIM...","CONTE MAIS...","CONT
     INUE...","E...","ENTAO..."
2250 DATA "ISTO E IMPORTANTE...","POR QUE
     ISTO E IMPORTANTE PARA VOCE ?","EXP
     LIQUE MELHOR"
2300 DATA "POR QUE VOCE PENSA ASSIM ?"
2350 DATA "COMO ISTO AFETA VOCE ?"
2500 FOR K=1 TO AN:READ R$(K):NEXT K
2600 HOME:O$="OLA! QUAL E' O PROBLEMA ?"
2950 RETURN
2995 REM ********************************
3000 REM **          ANALISA             **
3010 REM ********************************
3100 IF I$="ATE LOGO" THEN EX=L-1:L=2*LT
     :RETURN
3200 H$(L)=O$:H$(L+1)=I$
3300 R9=INT(AN*RND(1)+1):IF R9=R0 THEN 3
     300
3400 O$=R$(R9):R0=R9
3420 IF LEFT$(I$,3)="SIM" THEN O$="VOCE
     TEM CERTEZA MESMO ?":RETURN
3450 IF LEFT$(I$,3)= "NAO" THEN O$="POR
     QUE NAO ?":RETURN
3500 Z$=RIGHT$(I$,1)
3520 IF Z$="?" THEN O$= "POR QUE VOCE ME
     PERGUNTA ISTO ?"
```



e sim a incita com respostas como: "Mas é importante mesmo?" e "Fale mais sobre isso".

O programa mantém um registro da conversa e o mostra depois de certo número de sentenças, ou tão logo você digite "ATE LOGO".

Quando desenvolvido em direção à pseudo-inteligência, faz com que o computador analise as entradas do usuário e escolha a resposta apropriada. Em nosso exemplo, ele faz isso já na linha 3100, testando para ver se é "ATE LOGO".

Pode-se estender essa análise verificando se há as palavras "sim" e "não" e questioná-las. Alguma coisa simples como "POR QUE SIM?" ou "POR QUE NAO?" já faria efeito.

Dá bom resultado, também, conferir se o usuário está repetindo uma resposta e, se estiver, responder de acordo ou encerrar a sessão. Pode-se ainda testar se a resposta acaba em ponto de interrogação ou de exclamação, e responder: "POR QUE VOCE ME PERGUNTA ISTO?" ou "POR QUE ISTO O IMPORTUNA?". Essas estratégias são eficazes e não exigem muito tempo de processamento nem compreensão da resposta do usuário.

```
3550 IF Z$="!" THEN O$="POR QUE ISTO O IM
     PORTUNA ?"
3600 COM = LEN (I$)
3610 IF COM > 15 THEN GOSUB 4000
3950 RETURN
3990 REM ********************************
4000 REM **       FRASE MAIS LONGA       **
4010 REM ********************************
4050 W=1:H=1:WL=1:S=1
4100 I$=I$+" ":L9=LEN(I$)
4120 FOR C=1 TO L9:FOR P=C TO L9
4150 Z$=MID$(I$,P,1)
4170 IF Z$=" " THEN W=P-C:H=C:C=P:P=L9
4200 NEXT P
4220 IF W>WL THEN WL=W:S=H
4250 NEXT C
4270 O$="O QUE SIGNIFICA PARA VOCE " +
     CHR$(13) + "'" + MID$ (I$,S,WL) +
     "' ?": RETURN
4990 REM ********************************
5000 REM **          RELATORIO           **
5010 REM ********************************
5050 HOME:PRINT TAB(10);"RELATORIO"
5100 FOR K=1 TO EX STEP 2
5150 PRINT TAB(5);H$(K)
5200 PRINT H$(K+1)
5300 FOR D=1 TO T9:NEXT D
5400 NEXT K
5900 RETURN
```

Podemos, ainda, criar uma tabela de palavras-chaves e procurar respostas para elas. A escolha das palavras-chaves e das respostas depende do tipo de conversa que se espera ter.

O inconveniente desse método — e de todos os métodos de pesquisa de texto — é o tempo que demanda para analisar mesmo uma frase curta. A baixa velocidade do BASIC é o fator limitante; qualquer análise séria, no caso, requer um programa escrito em código de máquina.

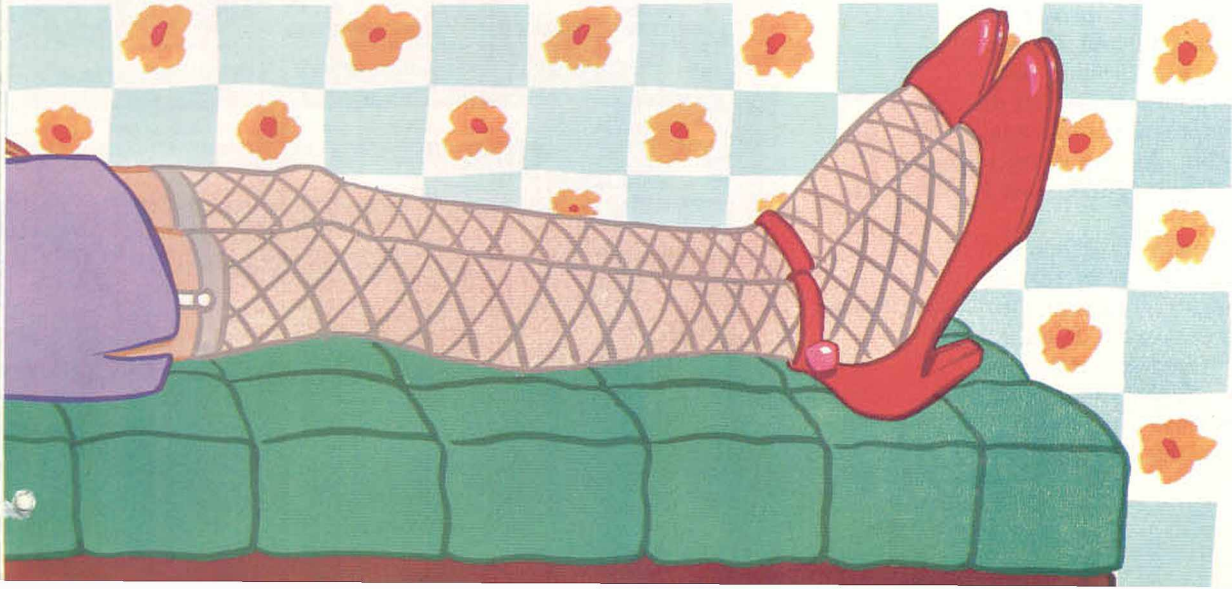
No entanto, essas demoras são perfeitamente toleráveis, e mesmo métodos simples podem levar a resultados surpreendentes, além de exemplificar os problemas que as máquinas de quinta geração terão de solucionar para compreender todas as sutilezas da linguagem humana, subjetiva e não padronizada.

Se pretendemos analisar as palavras do usuário, há muitas abordagens a serem seguidas. A mais interessante é a análise sintática — a redução de uma sentença a seus componentes, como sujeito, verbo e complemento. Isso exige um corpo de regras sintáticas e gramaticais e tabelas de pronomes, preposições, conjunções, flexões etc.

Seria mais fácil escolher as frases mais longas numa conversa banal e pedir ao usuário para explicar seus sentimentos sobre elas — essas frases tendem a ser importantes numa sentença simples. Os resultados melhoram quando se escolhem as palavras posteriores a "eu", "meu", "você" e "seu", por exemplo.

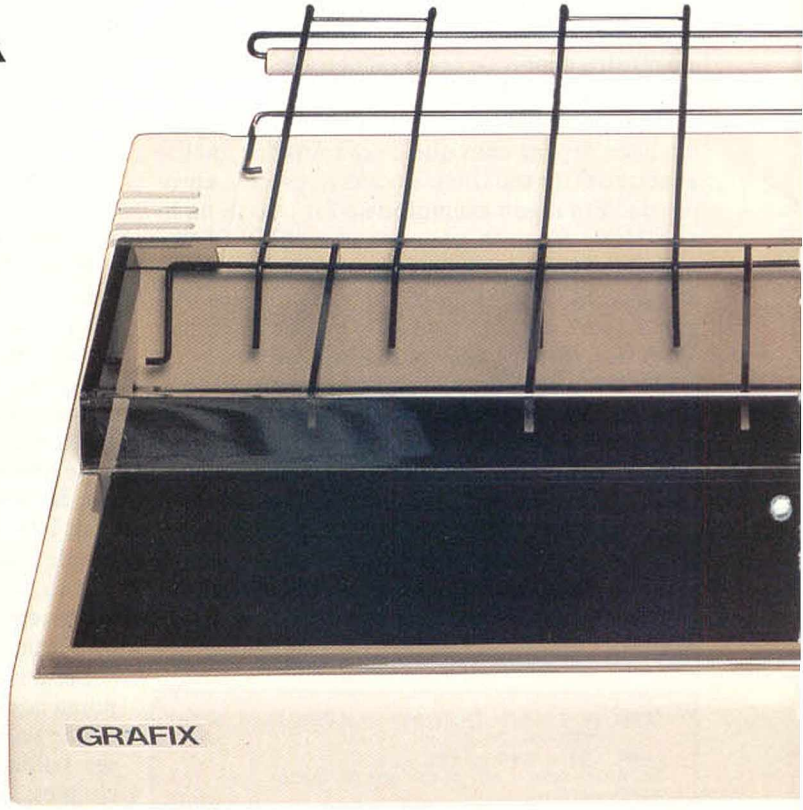
Escolher entre tais métodos e aperfeiçoá-los constitui fascinante exercício de programação. Você terá uma visão muito clara da imensa complexidade da linguagem humana e de sua análise e pode ficar insatisfeito com o BASIC como ferramenta de análise. Mas perceberá os aspectos positivos de linguagens de programação como LISP e PROLOG, estruturadas muito mais engenhosamente, sobretudo pela necessidade de lidar com as complexidades da linguagem — expressão do pensamento.

Além disso, você vai ter a oportunidade de conversar, pelo tempo que desejar, com o interlocutor perfeito — alguém que ouve.



GRAFIX 100MX

Amplos recursos para formatação de textos e variação de caracteres — além de uma grande resistência — são os maiores atrativos das impressoras matriciais Grafix MX para o usuário especializado.



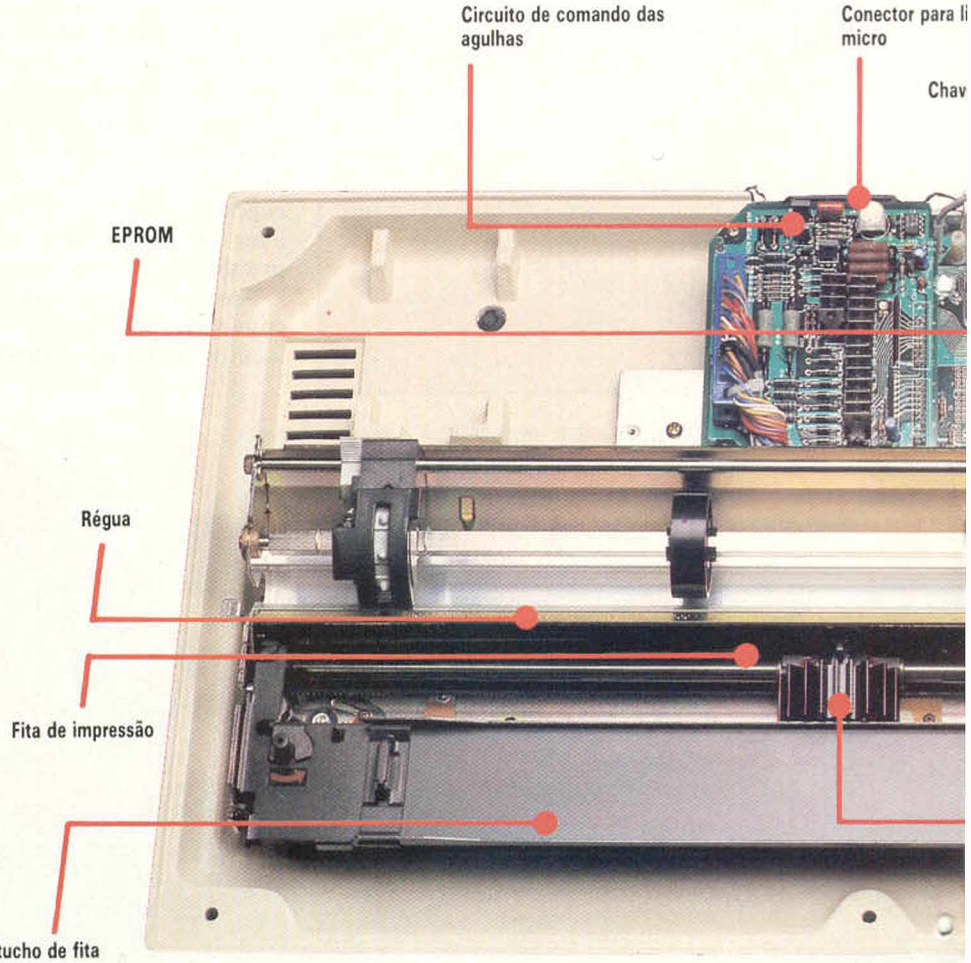
O usuário que precisa de uma impressora com recursos para listagem de programas, processamento de textos e produção de gráficos encontra nos modelos da linha Grafix MX, fabricados pela Scritta, uma boa opção entre as alternativas disponíveis no mercado nacional.

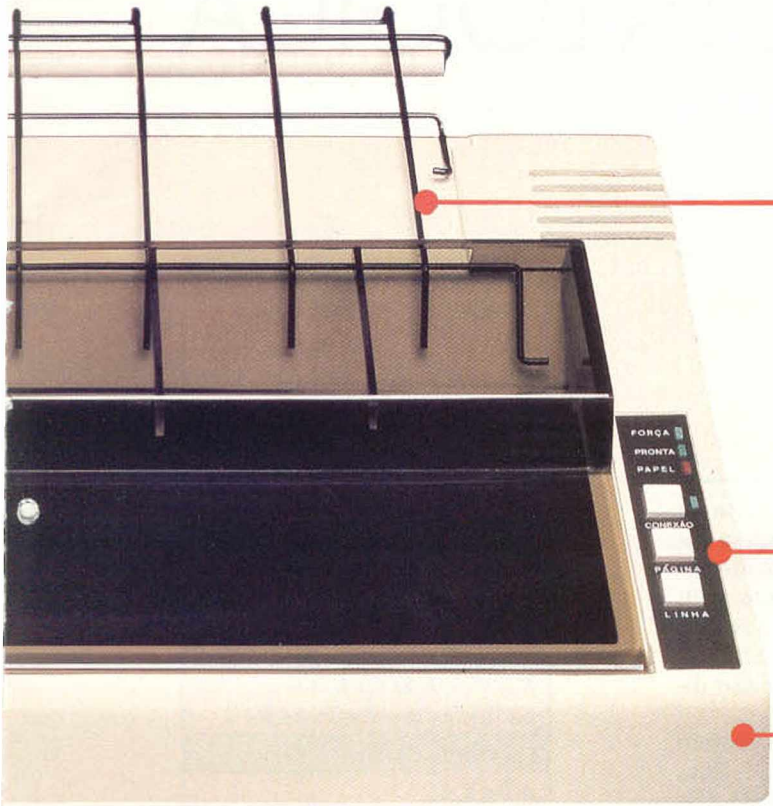
As impressoras matriciais Grafix 80MX e 100MX possuem praticamente as mesmas características, distinguindo-se apenas pela velocidade de impressão — 80 cps (caracteres por segundo) na primeira e 100 cps na segunda — e pela largura máxima dos formulários contínuos com que trabalham: de 4 a 10 polegadas e de 4 a 15 1/2 polegadas. Equipadas com a interface paralela Centronics ou, opcionalmente, com a RS232C ou a IEEE488, podem ser acopladas à maioria dos micros.

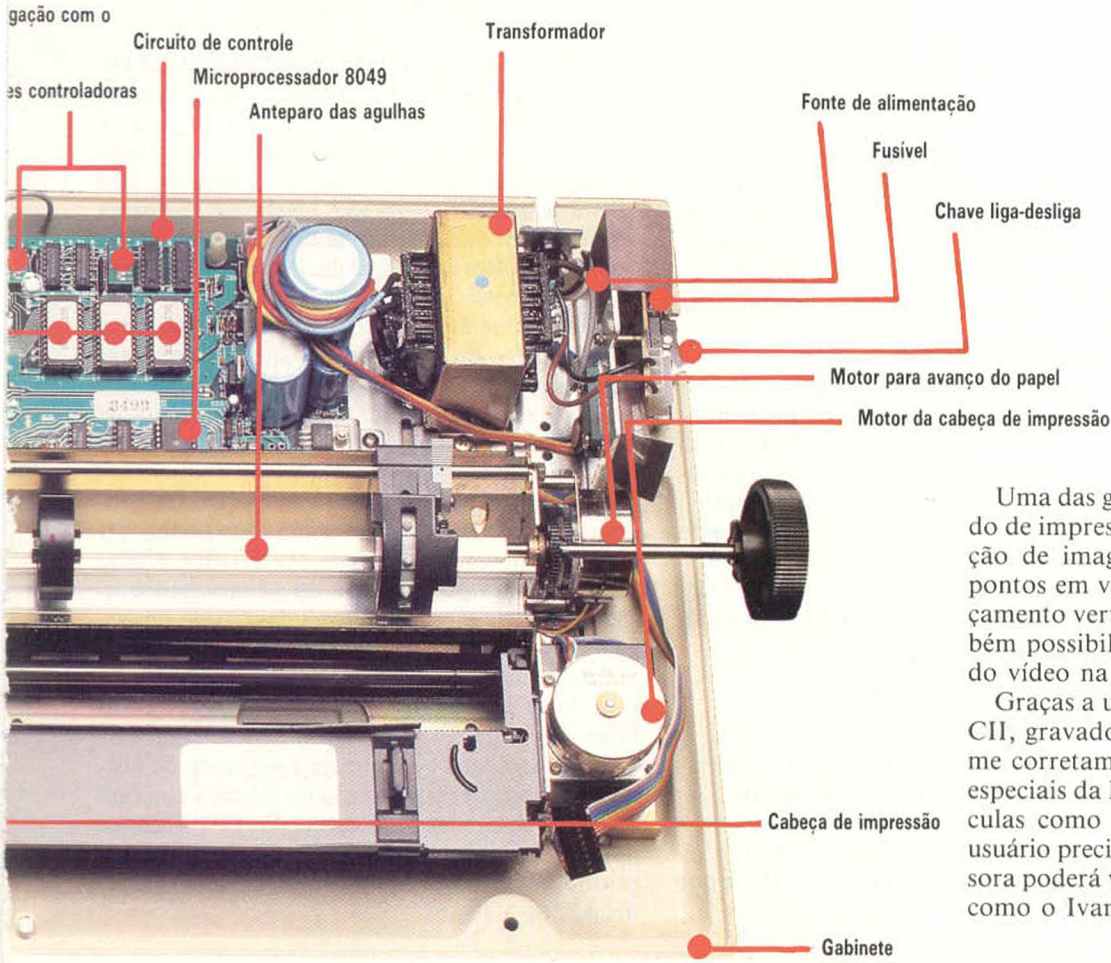
Utilizando matriz de 9 x 9 agulhas e imprimindo em ambas as direções (com dispositivo de pesquisa lógica), os dois modelos são bastante versáteis quanto aos recursos para apresentação do texto, possibilitando tarefas especializadas, tais como tabelas financeiras com muitas colunas, títulos em destaque etc. Para tanto, além do modo de impressão normal, com 10 cpp (caracteres por polegada), o usuário pode escolher entre os modos comprimido (17,16 cpp) e expandido (5 cpp, em que a largura dos caracteres é duplicada), ou, ainda, caracteres com largura dupla comprimida (8,5 cpp).

Quando se requer impressão com "qualidade de carta" — ou seja, semelhante à das máquinas de escrever —, a Grafix pode imprimir em negrito, rebatendo cada caractere com um deslocamento de 1/126 de polegada, fazendo desaparecer os espaços entre os pontos da matriz de agulhas; ou no modo de impressão enfatizada, que também pode ser combinado com o negrito, embora nesse caso a velocidade reduza-se à metade da normal.

Outros recursos incluem texto em itálico (em todas as opções de tamanho de caractere), sobrescrito e subscrito (uma letra ou qualquer outro caractere menor acima ou abaixo da linha), sublinhado; tabulação horizontal e vertical; entrelinhamento variável etc. Essas variações da escrita são ativadas, temporária ou permanentemente, por comandos introduzidos pelo software do micro.







GRAFIX 100MX

DIMENSÕES
592 x 393 x 133 mm.

PESO
10 kg.

MÉTODO DE IMPRESSÃO
Impacto por matriz de pontos.

VELOCIDADE DE IMPRESSÃO
100 cps.

DIREÇÃO DE IMPRESSÃO
Bidirecional, com pesquisa lógica.

MODOS DE IMPRESSÃO
Normal, duplo (negrito), enfatizado e duplo enfatizado.

AGULHAS DA CABEÇA
9.

MATRIZ
9 x 9.

TIPO DE PAPEL
Formulário contínuo.

LARGURA DO PAPEL
4 a 15 1/2".

NÚMERO DE CÓPIAS
Um original mais duas cópias com carbono.

COLUNAS
Normal, 136; dupla largura, 68; comprimida, 233; dupla largura comprimida, 116.

CONJUNTO DE CARACTERES
255 ASCII.

INTERFACE
Paralela, tipo Centronics.

DOCUMENTAÇÃO
Manual do usuário, didático e minucioso.

Uma das grandes vantagens da Grafix é o modo de impressão gráfica, que permite a elaboração de imagens de alta resolução, utilizando pontos em vez de caracteres e variando o espaçamento vertical e horizontal. Além disso, também possibilita o despejo (dump) do conteúdo do vídeo na impressora.

Graças a um repertório de 255 caracteres ASCII, gravado em três EPROM, a Grafix imprime corretamente todos os acentos e caracteres especiais da língua portuguesa, tanto em maiúsculas como em minúsculas. No entanto, se o usuário precisar de caracteres especiais, a impressora poderá vir equipada com outros repertórios como o Ivanita, o ABICOMP e o Europeu.

A RESPOSTA LÓGICA

Antes de passar a teorias mais adiantadas, veja um resumo das regras da álgebra booleana e uma série de exercícios para revisão dos princípios lógicos nela contidos.

Nos computadores, os princípios da lógica se aplicam ao projeto do hardware, destinado a desempenhar certas tarefas especiais. O circuito somador, por exemplo, ao ser combinado com outros circuitos somadores, possibilita a adição de números binários, imitando o método humano de adição. Isso permite o transporte dos dígitos de uma coluna para a seguinte.

Utilizamos três elementos básicos no projeto desse circuito, chamados "portas lógicas". Suas funções foram indicadas por E, OU e NÃO. Cada uma delas se definiu graças a uma tabela de validação — meio simples de escrever a(s) saída(s) de um circuito para qualquer combinação possível de entradas. Duas entradas nos deram quatro (2^2) combinações possíveis de entradas, três entradas nos deram oito (2^3) combinações e assim por diante.

Também já vimos como as portas lógicas são ligadas para a obtenção das saídas desejadas. Esses circuitos mais complexos poderiam também ser descritos por suas tabelas de validação. Em especial, projetamos um circuito OU-exclusivo, que resultava numa saída verdadeira quando somente *uma* de suas entradas fosse verdadeira.

A álgebra booleana

A combinação de elementos lógicos pode ser descrita no papel por um conjunto de símbolos muito semelhantes aos da álgebra normal. O ramo da matemática vinculado à representação da lógica é conhecido como álgebra booleana, homenagem ao matemático inglês George Boole (1815-1864), que pela primeira vez definiu e sistematizou seus princípios. Cada um desses elementos básicos tem seu próprio símbolo especial:

A E B	$A.B$
A OU B	$A+B$
NÃO A	$\overline{A}$

Assim como a álgebra comporta regras próprias para controlar a manipulação de números e letras, também existem leis especiais que regem a simplificação das expressões lógicas. As leis da álgebra booleana estão resumidas na tabela que se segue:

RELAÇÕES ESPECIAIS	
Relação	**Dual**
$A.A=A$	$A+A=A$
$A.\overline{A}=0$	$A+\overline{A}=1$
$A.0=0$	$A+1=1$
$A.1=A$	$A+0=A$
$A.(A+B)=A$	$A+A.B=A$
$A.(\overline{A}+B)=A.B$	$A+\overline{A}.B=A+B$
LEIS DE DE MORGAN	
1) $\overline{A+B}=\overline{A}.\overline{B}$	
2) $\overline{A.B}=\overline{A}+\overline{B}$	
LEI ASSOCIATIVA	
$A.(B.C)=(A.B).C=A.B.C$	
$A+(B+C)=(A+B)+C=A+B+C$	
LEI COMUTATIVA	
$A.B=B.A$	
$A+B=B+A$	
LEI DISTRIBUTIVA	
$A.(B+C)=A.B+A.C$	

O uso dessas regras torna possível simplificar expressões lógicas e reduzir o número de portas requeridas pelo circuito final. Mas, para a simplificação de circuitos, há também os mapas de Karnaugh, ou mapas-k, que, embora não substituam as simplificações algébricas, reduzem o esforço necessário para se lidar com a álgebra booleana. Esses mapas, extensões dos diagramas de Venn, possibilitam o agrupamento de expressões extraídas de uma tabela de validação para dois, quatros ou oitos (grupos que representam expressões booleanas mais simples). Na prática, muitas vezes se requer uma combinação de mapas-k com a álgebra para projetar o circuito mais eficiente.

Quanto às operações lógicas E e OU na programação em BASIC, esses comandos podem ser usados para combinar condições IF-THEN. Como já vimos, eles possibilitam ao programador ligar e desligar bits individuais dentro de um registro.

Finalizando o primeiro estágio do curso, sugeriu-se aplicar todo o conhecimento das tabelas de validação, dos mapas de Karnaugh, da álgebra booleana e da notação por meio de portas lógicas para projetar circuitos que fornecem as saídas desejadas para certas tarefas. Antes de lidar com teorias lógicas mais avançadas, é importante que você faça os exercícios de revisão propostos a seguir:

Exercício de revisão

1) Um conjunto de rock acredita no sucesso de sua mais recente gravação, desde que faça um bom videoclip *e* sua gravadora inclua um brinde no disco, *ou* desde que apareça num programa popular de tevê. Desenhe uma tabela de validação para todos os resultados possíveis. Se cada evento for igualmente provável, quais as possibilidades de o grupo vir a ter êxito?

2) O clube de xadrez de uma escola está admitindo novos sócios entre:

i) membros do corpo de funcionários;
ii) alunos do quarto *ou* quinto anos, que estudem matemática *e* ciências;
iii) alunos do sexto ano que estejam estudando matemática *ou* ciências.

Desenhe um emissor automático de carteiras de sócio que é operado pressionando-se os botões que melhor descrevem o candidato. Os botões são:

A = aluno do quarto ano;
B = aluno do quinto ano;
C = aluno do sexto ano;
D = membro do corpo de funcionários;
E = aluno estudante de matemática;
F = aluno estudante de ciências.

Faça um diagrama de circuitos para a máquina.

3) Certo microcomputador possui um registro com o endereço 23148 (decimal), utilizado para controlar a imagem no vídeo. O bit 0 é o menos significativo no registro e o bit 7 é o mais significativo. A tela pode ser passada para o modo de alta resolução colocando-se os bits 4 e 5 em um. Como os demais bits no registro são usados para controlar outras funções, é importante que seus valores não se alterem. Escreva comandos em BASIC que:

a) liguem a tela de alta resolução;
b) desliguem novamente a tela.

4) A caixa-forte de um banco funciona com um sistema de interruptores operados a chave. Três pessoas têm chave: o gerente, seu auxiliar e o tesoureiro. A porta se abre com duas das três chaves, mas:

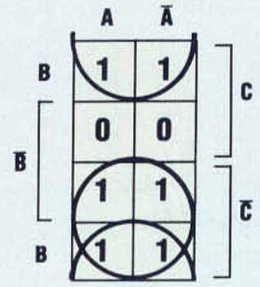
a) entre 9 e 17h, é indispensável a chave do tesoureiro;
b) nos outros horários, servem a do gerente e qualquer das duas restantes.

Desenhe um circuito para controlar o sistema.

5) Numa máquina de votar, cada bit de dados digitais binários é enviado simultaneamente para três canais paralelos que, na extremidade receptora, são fornecidos a um painel eletrônico. Não ocorrendo erros de transmissão, os três bits deverão ser idênticos. Caso contrário, o receptor de votos corrigirá o erro num canal dando uma saída simples, correspondente à maioria das entradas. Desenhe um circuito para o receptor.

Respostas do exercício 5

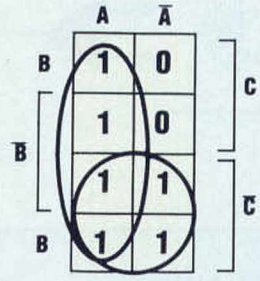
1a)



$= \bar{C} + B$
(alternativamente $= \overline{C.\bar{B}}$)

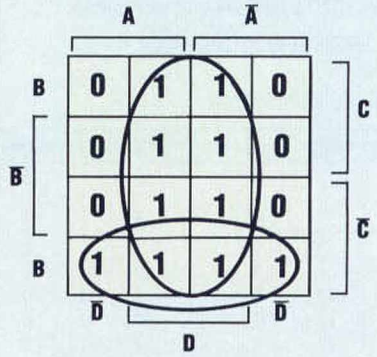
b)

$\overline{B + \bar{C}} + B.\bar{C} + A.C$
$= \bar{B}.\bar{C} + B.\bar{C} + A\,C$ (De Morgan)



$= A + \bar{C}$
(alternativamente $= \bar{A}.C$)

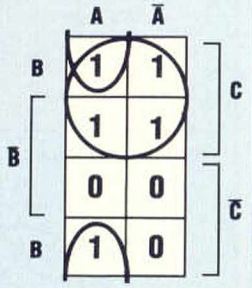
c)



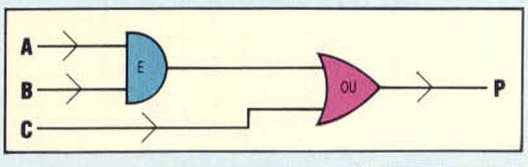
$= D + B.\bar{C}$

2) A tabela de validação é:

Decimal	A	B	C	P
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1



Do mapa-k obtém-se a expressão $P = C + A.B$. O circuito resultante é:



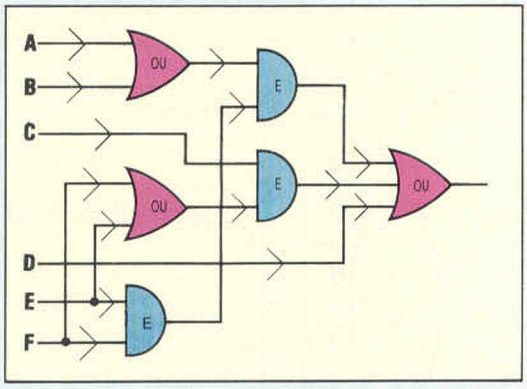
Resposta do exercício de revisão

1)

Bom vídeo	Brinde	Show na tevê	Êxito de vendas
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Probabilidade de êxito = 5/8 = 62,5%

2)



3 a) Para ligar a tela de alta resolução:

POKE23148,PEEK(23148)OU48

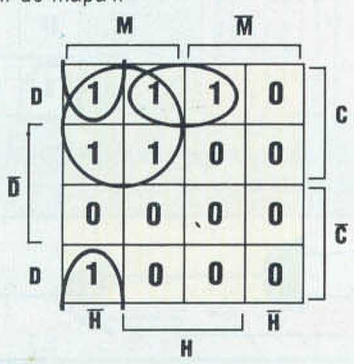
b) Para desligar a tela de alta resolução:

POKE23148,PEEK(23148)OU207

4) A tabela de validação é:

Gerente (M)	Auxiliar (D)	Tesoureiro (C)	Das 9 às 17h (H)	Abre a porta (U)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

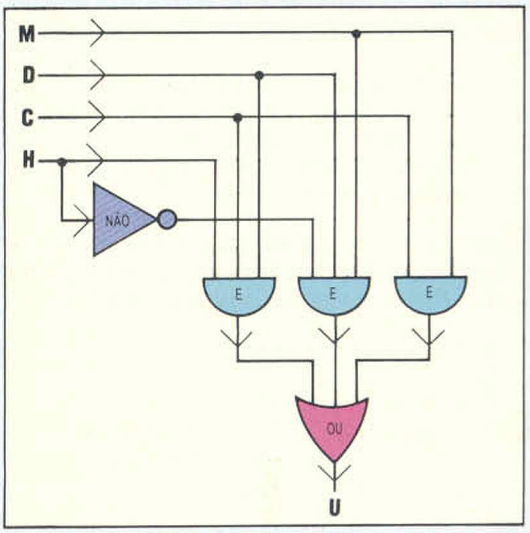
A partir do mapa-k



obtemos a expressão: $U = M.C + M.D.\overline{H} + D.C.H$. Para nos certificarmos de que não foi cometido algum erro, o melhor é reconverter essa expressão para frases em português. Segundo a expressão, a porta pode ser aberta por:

i) o gerente e o tesoureiro, a qualquer hora;
ii) o gerente e o auxiliar, fora do expediente; e
iii) o auxiliar e o tesoureiro, das 9 às 17h.

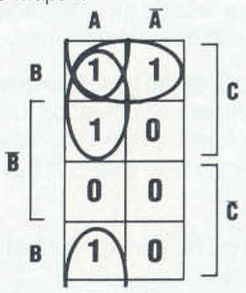
Eis o circuito correspondente:



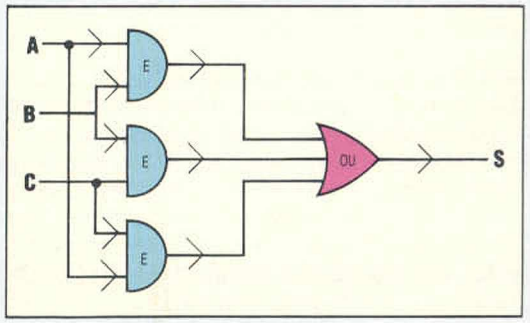
5) A tabela de validação é:

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Com a ajuda do mapa-k



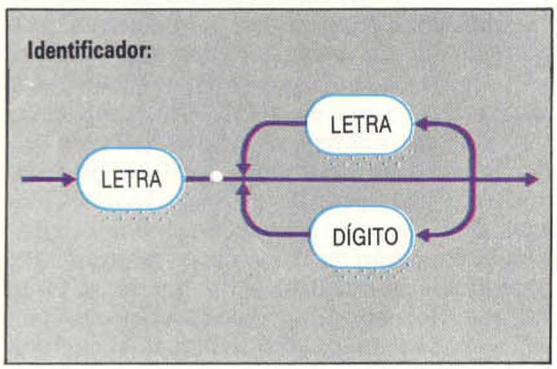
obtemos a expressão $S = A.B + A.C + B.C$, que corresponde a este circuito:



DADOS E DECISÕES

Quatro tipos de dados, instruções compostas e estruturas de tomada de decisões são os recursos de programação que examinaremos neste artigo da série sobre a poderosa linguagem PASCAL.

O PASCAL dispõe de quatro tipos simples de dados — ou seja, que podem ter apenas um valor —, representados pelos identificadores INTEGER, REAL, CHAR e BOOLEAN. Os números são classificados em fracionários (números reais) ou inteiros naturais (em inglês, integer). A faixa efetiva dos números disponíveis é limitada pela quantidade de bytes usada para armazenar determinado valor de cada um desses tipos. Cabe ao projetista do compilador (o "implementador") definir essas características, chamadas "dependentes da implementação". Tudo o que depende da implementação deve vir especificado na documentação segundo o padrão ISO (International Standards Organization) do PASCAL.



A faixa dos integers em seu compilador provavelmente se estende de −32.768 a 32.767 ou de −2.147.483.648 a 2.147.483.647, conforme seja usada representação de 2 ou de 4 bytes. No PASCAL representa-se o valor do número integer máximo pelo identificador de constante predefinido, MAXINT. Esse valor pode ser facilmente encontrado por meio da instrução:

```
WRITELN ('O MAXIMO INTEGER E:',MAXINT)
```

Os números reais também se restringirão a uma faixa e exatidão determinadas — em geral, de cerca de 1,7E+38 —, com seis ou sete dígitos de precisão, no mínimo. Essa forma de escrever os números reais, conhecida como "notação científica", é o padrão preestabelecido do PASCAL. Mas você pode usar o método mais comum de escrevê-los com uma vírgula decimal (por exemplo, 123,456).

CHAR é a abreviação de character (caractere), e um valor desse tipo será um dos elementos do conjunto de caracteres (geralmente ASCII) utilizado pelo computador. O PASCAL garante sua própria portabilidade ao supor que:

- Os caracteres de A a Z estarão ordenados alfabeticamente; isso significa que, em termos de valor de caractere, A será menor que B, este menor que C etc.
- Os caracteres numéricos de 0 a 9 estarão ordenados e contíguos; isto é, qualquer que seja o valor de 0, o 1 virá em seguida, e assim sucessivamente.

Cada valor de caractere terá um código numérico correspondente a um valor associado a uma subfaixa do tipo integer. Os códigos ASCII são definidos na faixa de 0 a 127, embora em muitas máquinas com códigos adicionais para caracteres gráficos esse limite seja ampliado para 225. Podemos estabelecer uma correspondência entre qualquer conjunto ordenado de caracteres e a escala de valores "ordinais" usada internamente pelo computador. Para isso, o PASCAL utiliza a função predefinida ORD: ela fornece o código integer de seu argumento — assim, ORD (A) equivale a 65 no conjunto de caracteres ASCII. Uma outra função, CHR, fornece o mapeamento no sentido inverso — CHR (65) gera o caractere A (note que o cifrão, $, *não* é usado com CHR).

A faixa de ambos — valores de caracteres e integers — é específica para cada equipamento e corresponde a uma escala ordenada de constantes conhecidas. Por esse motivo, são eles denominados "tipos ordinais" ou "escalares". Qualquer que seja o valor, sempre saberemos os valores seguintes, se existirem. Obtêm-se esses valores por meio de duas funções escalares:

```
PRED (ITEM) (PREDECESSOR)
SUCC (ITEM) (SUCESSOR)
```

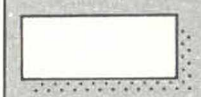
Assim, SUCC (3) corresponderá ao valor de caractere 4, mas PRED (Z) corresponderá apenas a Y em certos conjuntos de caracteres, como no ASCII, por exemplo. PRED (MAXINT) será 32.766 ou 2.147.483.646. Emprega-se a função CHR apenas com argumentos que sejam códigos de caracteres. Todas as outras funções escalares aceitam qualquer tipo escalar, embora ORD, ao ser usada com integers, forneça apenas o valor de seu argumento.

As variáveis booleanas são os tipos escalares mais simples de todos, pois há apenas dois valores na escala — falso (FALSE) e verdadeiro (TRUE). Assim, as funções escalares aplicam-se

Símbolos do PASCAL
Estas três formas são os símbolos mais usados nos diagramas de sintaxe:

- A de cantos arredondados ou em forma de losango representa as palavras reservadas, ou caracteres que não necessitam de outras explicações.

- Um círculo indica um operador do PASCAL (+, −, *, etc.).



- O retângulo indica uma palavra ou frase que possui seu diagrama de sintaxe separado.

a qualquer valor booleano — 0 é o valor ordinal de FALSE e 1 o de ORD(TRUE). As outras funções escalares, PRED e SUCC, não são muito úteis por enquanto.

O exemplo que segue de definição de constante de um programa mostra todos os tipos ordinais simples tal como aparecem no texto-fonte do PASCAL.

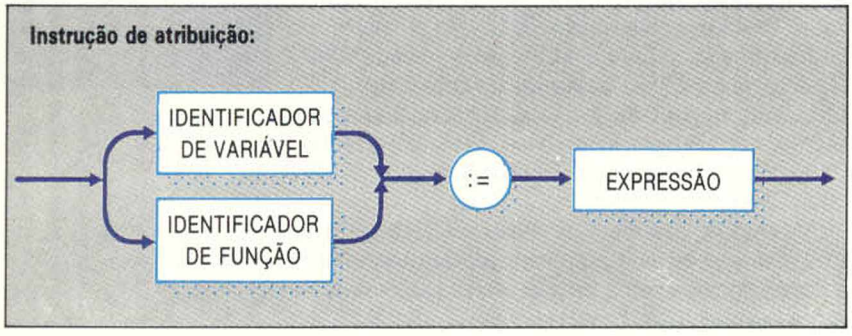
```
CONST
  VAT          = 0,15;
  COLUNAS      = 40;
  ESPACO       = ' ';
  DEBUGGING    = FALSE;
```

Igualdade e atribuição

O símbolo de igualdade (=) sempre significa "igual" em PASCAL e é usado para equiparar os identificadores de constante com seus valores. Quando introduzimos variáveis na seção de declaração VAR, o sinal de dois-pontos (:) separa o identificador de variável de seu tipo. Por exemplo:

```
VAR
  TAXA         : REAL;
  NUMERO       : INTEGER;
  SIMBOLO      : CHAR;
  FEITO        : BOOLEAN;
```

Para atribuir valores a essas variáveis, usa-se o "operador de atribuição" (:=) composto.



Isso ajuda a distinguir com clareza entre as três classes de operação. As definições CONST equalizam valores permanentes, as declarações VAR somente reservam espaço na memória, e ATRIBUICAO fornece um valor (possivelmente temporário) ao identificador.

Instruções compostas

Quando se executam duas ou mais instruções como parte de um único processo, elas são colocadas entre as palavras BEGIN e END, constituindo uma instrução "composta". Lembre-se de que cada instrução deve ser separada da seguinte por meio de ponto-e-vírgula. Já vimos uma instrução composta, uma vez que o núcleo de instruções executáveis de qualquer programa assume essa forma. Eis um programa completo que utiliza muitos dos recursos citados:

```
PROGRAM CIRCULO (INPUT, OUTPUT);
CONST
  PI           = 3.1415926536;
  QUESTAO      = 'ENTRE COM O RAIO:';
VAR
  RAIO,
  AREA         : REAL;
BEGIN
  WRITELN;
  WRITE (QUESTAO);
  READ (RAIO);
  AREA: = PI * RAIO * RAIO;
  WRITELN;
  WRITELN ('A AREA DO CIRCULO',
    'DE RAIO';RAIO : 8 : 3);
  WRITELN ('E :', AREA : 10 : 3);
END.
```

Em primeiro lugar, a parte VAR apresenta dois identificadores do mesmo tipo — ambos reais. Não são, no entanto, necessárias duas declarações separadas, pois uma vírgula separa as listas de itens. Do mesmo modo, quando especificamos mais de um argumento num procedimento — como nas instruções WRITELN —, a mesma regra se aplica. A segunda característica inovadora está na formatação da saída, que dá prioridade à notação científica dos valores reais. Temos a opção de especificar duas variáveis de integers separadas por sinais de dois-pontos, a fim de impor certo tamanho de campo para o número inteiro e sua parte fracionária.

Em nosso programa Círculo, atribuiremos três casas decimais para o raio e para a área. Como esta será um número maior (portanto mais longo), deixaremos um total de dez posições de caracteres para o raio, em vez de apenas oito. Esses valores integers devem ser maiores que zero, deixar espaço para um possível sinal, ter pelo menos um dígito e permitir a vírgula decimal colocada antes da parte fracionária. Valores ilegítimos ocasionarão erros quando o programa for processado, ou, na melhor das hipóteses, farão voltar à padronização original em notação científica; por exemplo, WRITELN (X : 6 : 2) não deixaria espaço para números maiores que 99,99.

Outro recurso ainda mais útil do PASCAL é o arredondamento automático do último dígito, possibilitando o máximo de precisão em qualquer campo numérico requisitado. Além disso, pode-se usar qualquer variável ou expressão, e não apenas uma constante. Isso permite enorme flexibilidade, inclusive recursos de tabulação. Com todos os demais tipos de dados, somente um valor integer é necessário — ou melhor, permitido — para especificar o tamanho do campo. Tal especificação fará com que os valores integers sejam escritos no campo de tamanho mínimo, sem espaços. Assim, teremos de nos lembrar de colocá-los nós mesmos, se quisermos a tabulação dos resultados. Por exemplo:

```
WRITELN ('TOTAL : ' : 20, PESO : 1, 'TONS.')
```

O PASCAL normalmente se recusará a fornecer um resultado impreciso, mas a supressão de to-

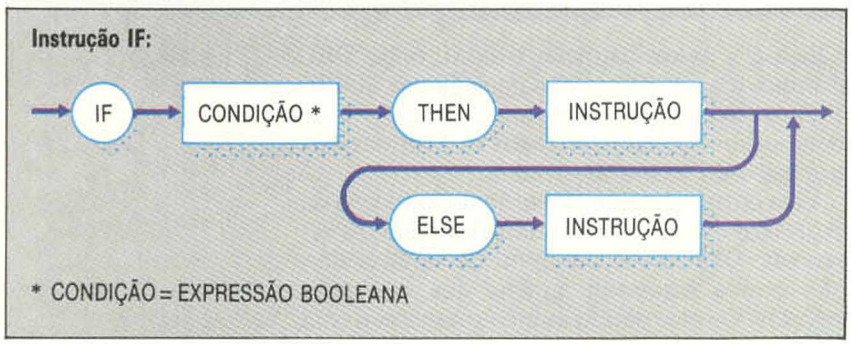
dos os espaços exige muito cuidado para que não sejam impressos dois números consecutivos num campo para um único número. Por exemplo, se 12 e 34 forem escritos dessa forma, o resultado será 1234.

Estruturas de decisão

A instrução IF do PASCAL é usada de modo semelhante à maioria das linguagens. Como o final de uma linha não encerra uma instrução, podemos usar a formatação livre do PASCAL para demonstrar as possíveis rotas através da estrutura da instrução IF. Eis duas dessas instruções:

```
IF CONTADOR = LIMITE
  THEN
    WRITELN ('SEM ESPACO')
  ELSE
    WRITE ('PROXIMO?');
IF NUMERO > MAXIMO THEN
  MAXIMO := NUMERO


ELSE
  {NENHUMA ACAO}
```

O alinhamento das palavras reservadas THEN e ELSE (quando presentes) facilita a visualização do fluxo de controle pela "estrutura". O ponto-e-vírgula é usado para separar instruções, como mostra o exemplo; portanto, *nunca* pode ser colocado antes de uma instrução ELSE. Lembre-se de que ELSE não é um comando em PASCAL, mas apenas IF; THEN e ELSE são usadas para delimitar a condição booleana e as instruções nas duas sentenças. Toda vez que usamos uma instrução IF, testamos um valor booleano. A "sentença" THEN só será executada se a expressão for verdadeira. Caso ela seja seguida por uma sentença ELSE, a instrução (ou instruções) contida nessa sentença será executada quando o valor booleano for falso.



O PASCAL foi a primeira linguagem a introduzir os dados conceituais conhecidos como "escalares numerados". Eles são úteis porque possibilitam uma visão abrangente da classificação dos dados, sem que precisemos traduzi-los mentalmente para códigos numéricos. Já vimos como, nesse caso, a simples definição de constante pode ser útil:

```
CONST
  LARGURA = 80;
```

O identificador de constante LARGURA pode então ser usado em todo o programa para referência ao número de colunas no sentido da largura da tela ou da impressora. Refazer o programa para um vídeo de 40 colunas torna-se assim simples questão de modificar a definição no início do programa. Uma escala completa de constantes define-se por meio desse método.

Se estivéssemos usando gráficos coloridos, as cores disponíveis seriam associadas a uma escala de números (vermelho = 1, verde = 2, por exemplo), o que — teoricamente — permitiria a extração da raiz quadrada de verde, ou a multiplicação de azul por amarelo. Isso, além de absurdo, é também uma fonte de erros em potencial quando estamos resolvendo problemas que não envolvem apenas dados numéricos. A parte da definição TYPE de um programa em PASCAL pode ser usada para definir um tipo escalar conceitual inteiramente novo, pela simples enumeração de uma lista de identificadores representando os valores constantes da escala. Por exemplo:

```
TYPE
  MATIZ = (VERMELHO, VERDE, AMARELO,
    AZUL);
```

Como se trata de um novo tipo de definição e não de uma declaração, a sintaxe utiliza o sinal de igualdade para definir o identificador de tipo (MATIZ) e indicar os valores ordenados do tipo entre parênteses. De modo análogo, o tipo predeterminado integer refere-se a todos os números integers existentes na implementação do PASCAL.

Como sempre acontece no PASCAL, os identificadores sucessivos numa lista (nesse caso, valores para cores) apresentam-se separados por vírgulas. Esses valores conceituais são automaticamente associados (pelo compilador) a valores integers, com a enumeração ordinal partindo de 0. Assim como o caractere A equivale a 65 no código ASCII, cada valor de MATIZ corresponde a um número ordinal, obtido por meio da função escalar ORD. No exemplo, ORD (VERMELHO) é 0, e ORD (AZUL) equivale a 3. Definido esse novo tipo escalar, declaramos uma variável de forma habitual:

```
VAR
  COR : MATIZ;
```

Essa estrutura declara que um identificador (COR) constitui um item de dados do tipo MATIZ, da mesma forma que a declaração

```
VAR
  LETRA : CHAR;
```

associa o tipo caractere ao objeto de dados chamado LETRA. As únicas operações definidas nos tipos enumerados são os testes de relação e o uso das funções escalares. Por exemplo, poderíamos escrever:

```
IF COR < AZUL THEN
  COR := SUCC (COR)
```

Lembre-se de que PRED (VERMELHO) e SUCC (AZUL) não existem. A variável COR é incompatível com variáveis de qualquer outro tipo, escalares ou não. Isso significa que não podemos executar funções ilegítimas como extrair a raiz quadrada ou dizer:

```
COR := COR + 1
```

Disso resulta uma limitação óbvia. Os caracteres e os números podem ser usados como parâmetros para instruções WRITE e WRITELN, mas

```
WRITELN (COR)
```

seria ilegítima, porque os valores do tipo são puramente conceituais e, se quisermos imprimir seus nomes, teremos de associar os valores de cores às cadeias de caracteres. Essa é uma utilização ideal para outra estrutura especial do PASCAL, a instrução CASE.

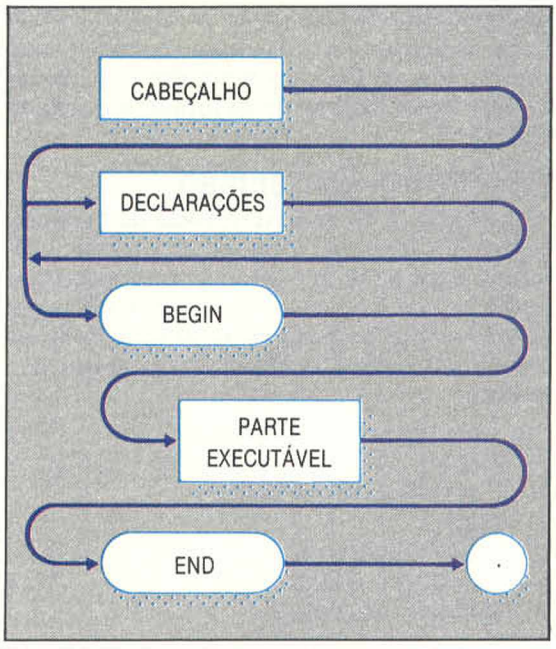
Vimos como, em PASCAL, a instrução IF parece bastante familiar e é facilmente visualizada graças às convenções de formatação livre. Contudo, há ocasiões em que devem ser tomadas decisões de opções múltiplas, que exigiriam algo do tipo:

```
IF N = 1
  THEN
    WRITE ('PRIMEIRO')
  ELSE
    IF N = 2
      THEN
        WRITE ('SEGUNDO')
      ELSE
        IF N = 3
          THEN
            WRITE ('TERCEIRO')
          ELSE
            WRITE ('QUARTO')
```

Sempre que a instrução a ser executada depender de o valor de um escalar simples estar num certo limite restrito, obteremos melhores resultados se empregarmos a instrução CASE. Nesse caso:

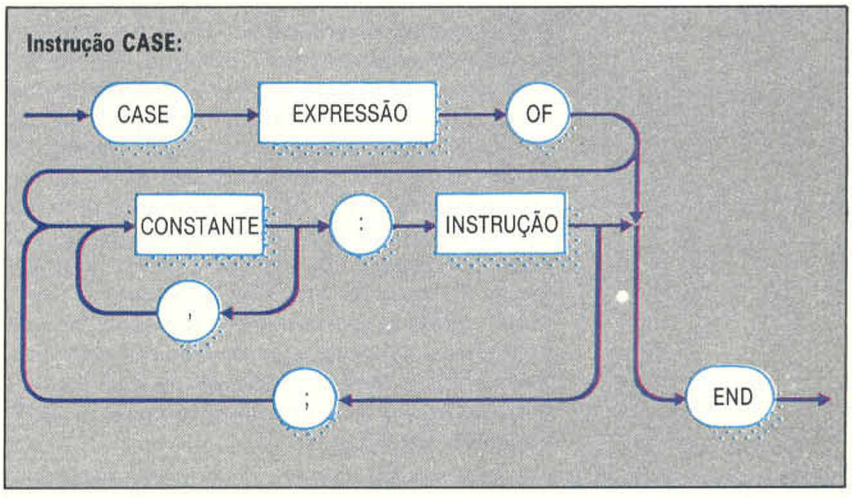
```
CASE N OF
  1          : WRITE ('PRIMEIRO');
  2          : WRITE ('SEGUNDO');
  3          : WRITE ('TERCEIRO');
  4,5,6
  7,8,9      : WRITE ('QUARTO')
END CASE
```

Note que isso será conveniente apenas para os valores de N na faixa dos "rótulos case" especificados no "núcleo" da instrução CASE — 1 a 9, no exemplo. Todos os valores que N poderá ter quando o programa for executado *deverão* estar relacionados individualmente, e não seria aceitável, por exemplo, introduzir N = 0.



Esquema do programa
Este é o diagrama básico da sintaxe de um programa em linguagem PASCAL, apresentando todos os seus elementos importantes. Os programas em PASCAL podem ser escritos com letras maiúsculas ou minúsculas; em algumas versões, apenas as palavras reservadas vêm em maiúsculas.

Muitos compiladores PASCAL também possuem a palavra reservada OTHERWISE ou OTHERS, que serve para relacionar um procedimento previamente padronizado.



Essa é a única instrução PASCAL onde END delimita uma estrutura não iniciada por BEGIN. Trata-se de um procedimento comum — e de bom estilo — qualificar todos os END das instruções CASE como mostramos.

Essa estrutura opera por meio da avaliação da expressão escalar delimitada pelas palavras reservadas CASE e OF. Um nome de variável simples é uma expressão mínima que não exige cálculos. O valor obtido é então comparado com cada constante relacionada nas listas de rótulo CASE e, ao ser encontrada uma correspondência, a instrução que vem após os dois-pontos — e somente essa instrução — é executada. Assim, o fluxo de controle não se "desvia", preservando a integridade estrutural. Se for necessária a execução de várias operações, emprega-se uma instrução composta, colocada entre o par BEGIN/END.

Em certas circunstâncias, alguns valores não requerem a execução de uma atividade, caso em que se pode usar a mais simples de todas as instruções do PASCAL a NULL, que significa "nada executar" e não envolve absolutamente sintaxe alguma. Eis um exemplo:

```
CASE N MOD 4 OF
  0 : {NADA EXECUTAR};
  1,3 : BEGIN
    WRITE (N MOD 4 : 1, 'QUARTO');
    IF N MOD 4>1 THEN
      WRITE ('S')
    END;
  2 : WRITE ('METADE')
END {CASE}
```

O ponto-e-vírgula ainda é necessário para separar essa instrução não existente daquela que vem a seguir, com exceção da última, pois ela vem antes de uma palavra reservada (END) e não de outro rótulo ou instrução. O uso do operador MOD na expressão garante que o valor esteja dentro do limite de 0 a 3. Ele fornece o resto de uma divisão integer, como acontece em várias versões do BASIC.

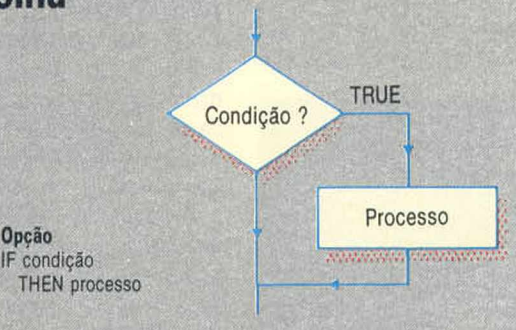
No próximo artigo examinaremos esse e os demais operadores do PASCAL, bem como as funções incorporadas. Por enquanto, aqui está a solução para o problema de como imprimir as cadeias de caracteres para cada valor de nosso próprio tipo de cor e matiz:

```
CASE COR OF
  VERMELHO  : WRITE ('VERMELHO');
  VERDE     : WRITE ('VERDE');
  AMARELO   : WRITE ('AMARELO');
  AZUL      : WRITE ('AZUL');
END {CASE}
```

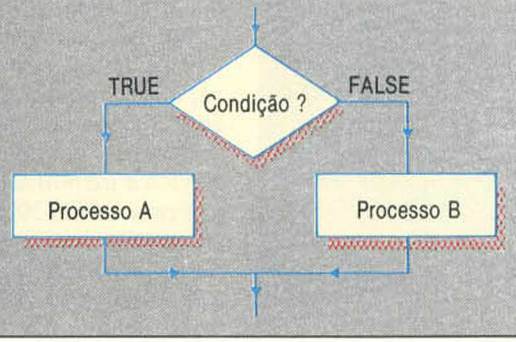
Estrutura de escolha

A estrutura IF-THEN executará o "processo" se a condição for verdadeira. Caso seja falsa, o programa simplesmente se desviará para a instrução seguinte.

Enquanto a estrutura IF-THEN apresenta uma "opção", em IF-THEN-ELSE ocorre uma "escolha". Conforme o resultado do teste realizado no início da estrutura, um dos dois processos será executado.



Opção
IF condição
THEN processo



Escolha
IF condição
THEN processo A
ELSE processo B

Instruções para o mês

Este programa, que calcula o número de dias entre uma data fornecida pelo usuário e o final do mês, apresenta alguns dos principais recursos da programação em PASCAL.

A instrução CASE é usada para associar os dados numéricos de entrada aos tipos numerados do PASCAL, usando-os para o processamento e depois associando-os de novo à cadeia de dados de saída. Note o uso da variável CHAR para ler qualquer caractere que separe os números introduzidos.

```
PROGRAM     DATA     (INPUT, OUTPUT);

CONST
  PONTO FINAL = '.';

TYPE

  CALENDARIO = (JAN, FEV, MAR, ABR, MAI,
  JUN, JUL, AGO, SET, OUT, NOV, DEZ);

VAR
  NOMEMES          : CALENDARIO;
  DIA,
  MES,
  ANO,
  FALTAM           : INTEGER;
  SIMBOLO          : CHAR;
  BISSEXTO         : BOOLEAN;


BEGIN
  WRITELN ('ENTRE COM A DATA NA FORMA :');
  WRITELN ('DD/MM/AA' : 40);
  WRITELN;
  WRITE ('DATA = >');
  READ (DIA, SIMBOLO, MES, SIMBOLO, ANO);
  BISSEXTO := ANO MOD 4 = 0;
  IF (MES >0) AND (MES < = 12)
    THEN
      CASE MES OF

        1          : NOMEMES := JAN;
        2          : NOMEMES := FEV;
        3          : NOMEMES := MAR;
        4          : NOMEMES := ABR;
        5          : NOMEMES := MAI;
        6          : NOMEMES := JUN;
        7          : NOMEMES := JUL;
        8          : NOMEMES := AGO;
        9          : NOMEMES := SET;
        10         : NOMEMES := OUT;
        11         : NOMEMES := NOV;
        12         : NOMEMES := DEZ;

      END;         {CASE}

    ELSE
      BEGIN
        WRITELN  ('EPA!');
        WRITELN  ('O PROGRAMA ASSIM');
        WRITELN  ('NAO FUNCIONA!');
      END;


    CASE NOMEMES OF
    JAN, MAR,
    MAI, JUL       : FALTAM := 31-DIA;
    ABR, JUN,
    SET, NOV       : FALTAM := 30-DIA;
    FEV            : IF BISSEXTO
                       THEN
                         FALTAM := 29-DIA;
                       ELSE
                         FALTAM := 28-DIA;


  END; {CASE}
  WRITELN ('FALTAM', FALTAM : 1,
      'DIAS PARA TERMINAR');

      CASE NOMEMES OF
      JAN          : WRITE ('JANEIRO');
      FEV          : WRITE ('FEVEREIRO');
      MAR          : WRITE ('MARCO');
      ABR          : WRITE ('ABRIL');
      MAI          : WRITE ('MAIO');
      JUN          : WRITE ('JUNHO');
      JUL          : WRITE ('JULHO');
      AGO          : WRITE ('AGOSTO');
      SET          : WRITE ('SETEMBRO');
      OUT          : WRITE ('OUTUBRO');
      NOV          : WRITE ('NOVEMBRO');
      DEZ          : WRITE ('DEZEMBRO');


      END; {CASE}
    WRITELN (PONTO FINAL);
  END.
```

ELEGÂNCIA ITALIANA

No mercado competitivo dos equipamentos para escritório, uma multinacional com base européia, a Olivetti, conseguiu se impor como fabricante de máquinas elegantes e de qualidade.

O M20
O computador M20, máquina de 16 bits, foi lançado no mercado em 1981. Incorpora um microprocessador Zilog Z8000 e usa o sistema operacional PCOS da Olivetti.

Camillo Olivetti fundou sua empresa em 1908. Com vinte empregados, montou uma pequena fábrica em Ivrea, no norte da Itália, e deu início à produção de sua primeira máquina de escrever, a M1.

Embora nessa época a economia italiana fosse predominantemente agrícola, a Olivetti cresceu, passando de quatro máquinas produzidas por dia, em 1914, para cinqüenta, em 1929.

Na década de 30, Adriano Olivetti, filho de Camillo, começou a reestruturar a companhia, empregando alunos da escola noturna da própria Olivetti, fundada em 1924. Foram também construídas moradias para os empregados. A empresa preocupava-se com a promoção de benefícios sociais e aplicou uma política funcional de assistência ao trabalhador.

Enquanto as indústrias do resto do mundo lutavam para sobreviver — o período era de depressão econômica em todo o mundo —, a Olivetti continuava crescendo. Em 1933, havia vendido aproximadamente 15 milhões de produtos para escritório. Quatro anos depois, lançou com êxito sua primeira teleimpressora, seguida, em 1940, pela primeira calculadora.

A Segunda Guerra Mundial (1939-1945) trouxe uma estagnação temporária, mas no período posterior ao conflito a empresa concentrou-se no desenvolvimento de novos mercados internacionais, com um êxito que teve como fundamentos principais a qualidade dos produtos e seu design elegante.

Nas décadas de 50 e 60, a Olivetti voltou-se para os computadores. Esse processo começou em 1955, com uma máquina para cálculos. Seu primeiro computador de grande porte, o Elea, foi desenvolvido alguns anos depois.

A empresa continuou a diversificar, produzindo também aparelhos eletrônicos. Uma nova linha de minicomputadores foi lançada, juntamente com terminais bancários e equipamentos de comunicação.

Atualmente, a Olivetti fabrica grande variedade de equipamentos profissionais para escritório e investe somas consideráveis no desenvolvimento de software apropriado para suas máquinas. Em 1982, foi a segunda maior indústria de computadores da Europa (só ultrapassada pela IBM), vendendo bem tanto seu portátil M10 quanto o computador profissional M20.

Na ocasião, a empresa italiana já planejava lançar uma máquina compatível com o IBM PC, mas de custo inferior. Denominada M24, teria como características um processador 8086-2 e a opção de aceitar uma placa Z8001 para torná-la compatível com o M20 da própria Olivetti. Em função disso, porém, a indústria precisou abandonar seu sistema operacional PCOS.

Arquitetura compatível
A Olivetti sempre foi conhecida pelo avançado design de seus produtos, conceito que se estende também à arquitetura dos edifícios da empresa. As instalações da foto são da Olivetti brasileira.

Empreendimentos brasileiros

A Olivetti instalou-se no Brasil em 10 de novembro de 1952. Sete anos depois, inaugurou sua fábrica em Guarulhos (SP) e iniciou a comercialização de máquinas de escrever, de calcular (mecânicas e eletromecânicas) e de contabilizar, assim como as de telex.

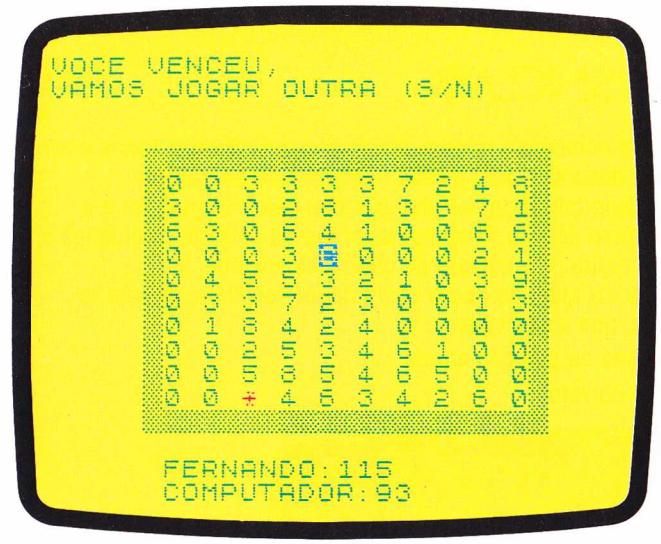
A década de 60 caracterizou-se, para a Olivetti brasileira, pelos equipamentos eletrônicos. Os primeiros foram as máquinas contábeis e as centrais de telecomunicação, seguindo-se calculadoras eletrônicas programáveis, terminais para teleprocessamento e fotocopiadoras.

Em 1985, eram três os produtos eletrônicos da Olivetti que se destacavam no mercado brasileiro: a máquina de escrever ET 121, do tipo margarida, que se transforma numa impressora acoplável a um micro, com seletores e teclas de serviço e memória de uma linha; o Sistema Data Entry SDE-2500; e a teleimpressora TE 520. Na mesma época, a empresa planejava lançar a máquina de escrever eletrônica portátil Praxis 20, que vinha sendo fabricada exclusivamente para exportação.

```
 480 PRINT AT 14,0;D$
 490 IF COMP>7.5 THEN GOTO 690
 500 IF F1=1 THEN GOTO 540
 510 IF COMP>=5 AND COMP<=6 THEN
 LET F1=1
 520 IF COMP>6 THEN GOTO 540
 530 GOTO 460
 540 IF NOT (COMP>=CF AND COMP<=
7.5 AND CF<=7.5) THEN GOTO 570
 550 LET CC=CC+5
 560 GOTO 850
 570 IF NOT (COMP<=7.5 AND CF>7.
5) THEN GOTO 610
 580 LET CP=CP-5
 590 LET CC=CC+2*5
 600 GOTO 850
 610 IF NOT (COMP<=CF AND COMP<=
7.5 AND CF<=7.5) THEN GOTO 650
 620 LET CP=CP+2*5
 630 LET CC=CC-5
 640 GOTO 850
 650 IF NOT (COMP>7.5 AND CF>7.5
) THEN GOTO 680
 660 LET CP=CP+3*5
 670 LET CC=CC-2*5
 680 GOTO 850
 690 IF NOT (CF>7.5) THEN GOTO 7
30
 700 LET CC=CC-2*5
 710 LET CP=CP+3*5
 720 GOTO 850
 730 LET CC=CC-5
 740 LET CP=CP+2*5
 750 GOTO 850
 760 IF NOT (CF>7.5) THEN GOTO 8
00
 770 LET CC=CC+2*5
 780 LET CP=CP-5
 790 GOTO 850
 800 LET CC=CC+5
 810 GOTO 850
 820 PAUSE 100
 830 IF P$="3" THEN GOTO 310
 840 GOTO 760
 850 PAUSE 200
 860 CLS
 870 PRINT "CAIXA=";CC;" CR$";AT
 1,0;"VOCE=";CP;" CR$"
 880 PAUSE 200
 890 CLS
 900 IF CC<=0 THEN GOTO 940
 910 IF CP>0 THEN GOTO 120
 920 LET W$="VOCE ESTA"
 930 GOTO 950
 940 LET W$="EU ESTOU"
 950 PRINT W$;" SEM DINHEIRO","Q
UER JOGAR OUTRA ? (S/N)"
 960 INPUT R$
 970 IF R$="N" THEN STOP
 980 RUN
 990 FAST
1000 DIM B(13,2)
1010 FOR F=1 TO 13
1020 LET B(F,2)=0
1030 LET B(F,1)=F
1040 IF F>10 THEN LET B(F,1)=.5
1050 NEXT F
1060 PRINT AT 9,0;"
"
1070 PRINT AT 13,0;"
"
1080 RETURN
1090 IF P$="3" OR P$="4" THEN GO
TO 820
1100 LET I=INT (RND*13)+1
1110 IF I>7 AND I<11 THEN GOTO 1
100
1120 IF B(I,2)>=4 THEN GOSUB 110
0
1130 LET B(I,2)=B(I,2)+1
1140 LET C$=C$+Z$(I)
1150 LET D$=D$+Z$(I)
1160 IF BAN=1 THEN RETURN
1170 IF P$="2" THEN GOTO 1210
1180 LET CA=CA+B(I,1)
1190 LET CF=CF+B(I,1)
1200 GOTO 1240
1210 LET CA=CF
1220 LET CF=CA+B(I,1)
1230 LET D$=C$
1240 RETURN
1250 LET P$="2"
1260 GOSUB 1090
1270 LET D$=CHR$ (CODE D$+128)
1280 PRINT AT 10,0;D$
1290 PRINT AT 7,0;"APOSTA ?"
1300 INPUT S
1310 IF S>CP THEN LET S=CP
1320 LET CP=CP-S
1330 PRINT AT 7,6;"=";S;" CR$"
1340 RETURN
```

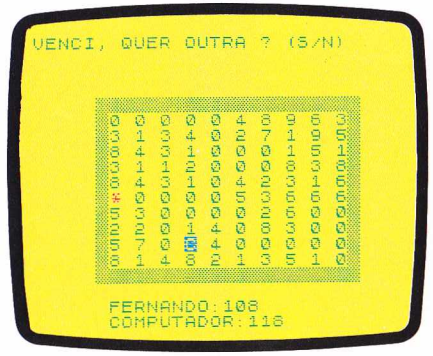
110
PARA A LINHA SINCLAIR

Rápido e simples, este jogo vai exigir de você o máximo de velocidade, pois seu adversário será a máquina. Veja se consegue atingir (ou ultrapassar) 110 pontos mais depressa que o computador. Experimente. Para tanto, jogue controlando o asterisco (que indica o número escolhido), por meio das teclas [5], [6], [7] e [8]. Os lances se alternam: o micro o convida para iniciar e joga em seguida. Os números escolhidos são substituídos na tela pelo zero, enquanto no rodapé se vê a contagem de cada um. Nas linhas 190, 200 e 220, o caractere gráfico é o da tecla [A].



Percurso da vitória
Você joga controlando o asterisco, enquanto o computador move o C na tela. Escolha o percurso capaz de lhe proporcionar maior número de pontos em menos tempo.

```
  10 REM
  20 REM          110
  30 REM  PARA A LINHA SINCLAIR
  40 REM
  50 PRINT "QUAL O SEU NOME ?"
  60 INPUT N$
  70 PRINT N$;" VOCE COMECA JOGA
NDO"
  80 PAUSE 300
  90 CLS
 100 FAST
 110 LET PP=0
 120 LET PPC=0
 130 LET DF=1+PEEK 16396+256*PEE
K 16397
 140 FOR L=6 TO 24 STEP 2
 150 FOR C=5 TO 14
 160 PRINT AT C,L;CHR$ (29+INT (
RND*9))
 170 NEXT C
 180 NEXT L
 190 PRINT AT 4,5;"
"
 200 PRINT AT 15,5;"
"
 210 FOR F=5 TO 14
 220 PRINT AT F,5;"";AT F,25;"
"
 230 NEXT F
 240 SLOW
 250 LET P=DF+171
 260 POKE P,23
 270 LET PC=DF+486
 280 POKE PC,168
 290 IF INKEY$="" THEN GOTO 290
 300 LET A$=INKEY$
 310 IF A$="5" THEN LET R=-2
 320 IF A$="6" THEN LET R=33
 330 IF A$="7" THEN LET R=-33
 340 IF A$="8" THEN LET R=2
 350 IF PEEK (P+R)<28 OR PEEK (P
+R)>37 THEN GOTO 290
 360 POKE P,28
 370 LET P=P+R
 380 LET PP=PP+PEEK P-28
 390 POKE P,23
 400 PRINT AT 17,6;N$;":";PP
 410 LET X1=PEEK (PC-33)
 420 LET X2=PEEK (PC+33)
 430 LET X3=PEEK (PC+2)
 440 LET X4=PEEK (PC-2)
 450 LET PC1=PC
 460 IF NOT (X1>=X2 AND X1>=X3 A
ND X1>=X4) THEN GOTO 490
```




```
 470 LET PC=PC-33
 480 GOTO 560
 490 IF NOT (X2>=X1 AND X2>=X3 A
ND X2>=X4) THEN GOTO 520
 500 LET PC=PC+33
 510 GOTO 560
 520 IF NOT (X3>=X4 AND X3>=X1 A
ND X3>=X2) THEN GOTO 550
 530 LET PC=PC+2
 540 GOTO 560
 550 LET PC=PC-2
 560 LET PPC=PPC+PEEK PC-28
 570 POKE PC1,28
 580 POKE PC,168
 590 PRINT AT 18,6;"COMPUTADOR:"
;PPC
 600 IF PP>=110 THEN GOTO 630
 610 IF PPC>=110 THEN GOTO 650
 620 GOTO 290
 630 PRINT AT 0,0;"VOCE VENCEU,"
,,"VAMOS JOGAR OUTRA (S/N)"
 640 GOTO 660
 650 PRINT AT 0,0;"VENCI, QUER O
UTRA ? (S/N)"
 660 INPUT X$
 670 IF X$="S" THEN GOTO 90
```

SENHA
PARA A LINHA SINCLAIR

O computador escolhe um número de quatro algarismos, no máximo, todos entre zero e cinco, e você deve descobrir qual é esse número. Não se trata de adivinhação. Será necessário investigar as pistas dadas pelo computador e a partir delas deduzir o número correto. O computador lhe dá seis oportunidades e, em cada uma, você deve digitar um número de quatro algarismos.
Se o número digitado não for igual ao que o computador escolheu, oriente-se pelas letras P e B na tela, que significam:

B, algarismo correto, mas na posição errada;

P, algarismo e posição corretos.

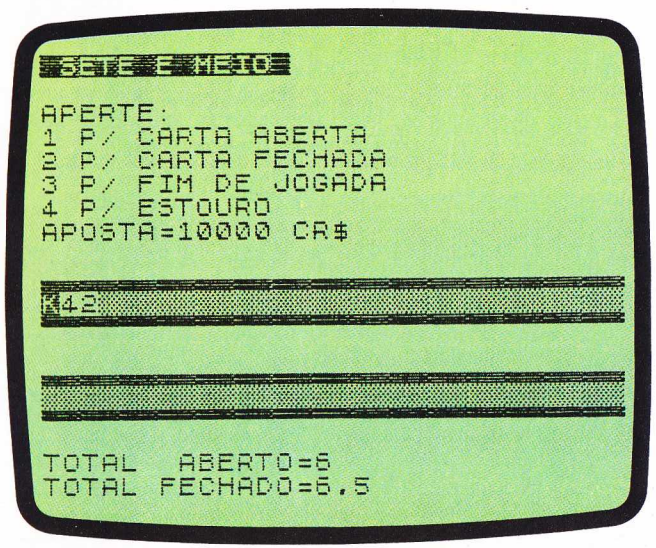
Dedução pura
Com seis oportunidades e algumas pistas, você pode descobrir o número escolhido pelo computador.

Algarismos repetidos
Você deve digitar quatro algarismos, sempre entre zero e cinco. O número da senha pode repetir todos os algarismos.

```
  10 REM
  20 REM      SENHA
  30 REM  PARA A LINHA SINCLAIR
  40 REM
  50 PRINT AT 0,12;"SENHA"
  60 DIM A$(3,4)
  70 LET C=0
  80 RAND
  90 FOR X=1 TO 4
 100 LET A$(1,X)=CHR$ INT (RND*5
+28)
 110 NEXT X
 120 PRINT AT 4,0;
 130 LET C=C+1
 140 IF C=7 THEN GOTO 375
 150 INPUT A$(2)
 160 IF A$(2)="    " THEN GOTO 3
75
 170 PRINT TAB 12;A$(2);" ";
 180 IF A$(2)=A$(1) THEN GOTO 37
0
 190 LET A$(3)=A$(1)
 200 FOR X=1 TO 4
 210 IF A$(2,X)<>A$(3,X) THEN GO
TO 250
 220 LET A$(3,X)="P"
 230 PRINT "P";
 240 LET A$(2,X)="/"
 250 NEXT X
 260 FOR X=1 TO 4
 270 IF A$(3,X)="P" THEN GOTO 34
0
 280 FOR Y=1 TO 4
 290 IF X=Y OR A$(3,X)<>A$(2,Y)
THEN GOTO 330
 300 LET A$(2,Y)="?"
 310 PRINT "B";
 320 GOTO 340
 330 NEXT Y
 340 NEXT X
 350 PRINT
 360 GOTO 130
 370 PRINT ,,"PARABENS"
 375 PRINT "O NUMERO ERA ";A$(1)
 380 PRINT AT 21,0;"DESEJA JOGAR
NOVAMENTE ? (S/N)"
 390 IF INKEY$="S" THEN GOTO 420
 400 IF INKEY$="N" THEN GOTO 450
 410 GOTO 390
 430 CLS
 440 RUN
 450 CLS
 460 PRINT AT 10,13;"FIM"
```

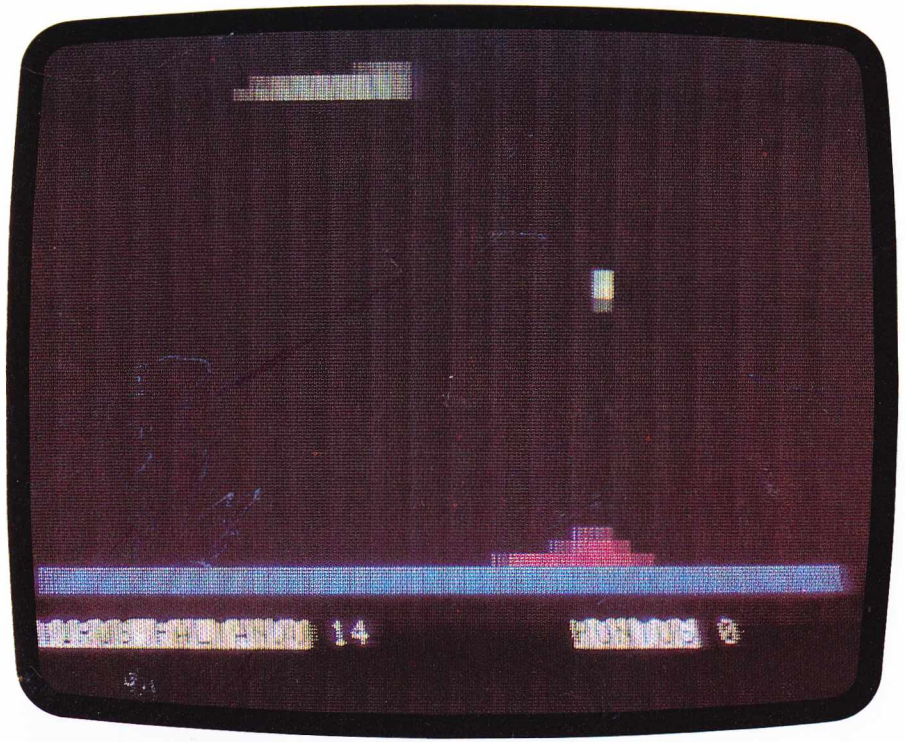

SETE E MEIO
PARA A LINHA SINCLAIR

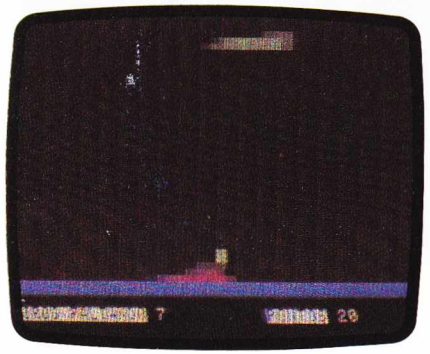
Este programa é uma adaptação do clássico jogo de cartas sete-e-meio, com uma diferença: não dá para blefar. O computador faz a "banca" e ambos começam com dez mil cruzeiros. Você deve pedir cartas de modo a totalizar um máximo de sete pontos e meio (figura vale meio ponto; as demais cartas, o valor que indicam). Ao terminar de pedir cartas, ganha quem tiver mais pontos.
Se você ultrapassar sete e meio, perderá, pois terá "estourado".
Faça as jogadas de acordo com as opções sugeridas no vídeo.



Quem tem mais sorte?
Você vai jogar contra o computador, com boas possibilidades de ganhar, pois este é um jogo de azar. Na tela, você encontra as instruções para pedir cartas. Ganha quem fizer mais pontos.

```
  10 REM
  20 REM      SETE E MEIO
  30 REM  PARA A LINHA SINCLAIR
  40 REM
  50 RAND
  60 CLS
  70 PRINT "VOCE E O COMPUTADOR
(CAIXA)","COMECAM COM CR$ 10.000
,00",,,"BOA SORTE"
  80 PAUSE 300
  90 LET CC=10000
 100 LET CP=10000
 110 CLS
 120 LET Z$="A234567---JQK"
 130 LET COMP=0
 140 LET CA=0
 150 LET CF=0
 160 LET C$=""
 170 LET D$=""
 180 LET BAN=0
 190 PRINT AT 0,0;"SETE E MEIO
"
 200 PRINT ,,"DIGITE:",,"1 P/ CA
RTA ABERTA","2 P/ CARTA FECHADA"
,"3 P/ FIM DE JOGADA","4 P/ ESTO
URO"
 210 GOSUB 990
 220 IF D$="" THEN GOSUB 1250
 230 INPUT P$
 240 IF P$>"4" OR P$<"1" THEN GO
TO 230
 250 LET P$=P$(1)
 260 GOSUB 1090
 270 IF P$="2" THEN LET D$(LEN D
$)=CHR$ ((CODE C$(LEN C$))+128)
 280 PRINT AT 10,0;D$
 290 PRINT AT 18,0;"TOTAL  ABERT
O=";CA;"     ","TOTAL FECHADO=";C
F;"  "
 300 GOTO 230
 310 LET F1=0
 320 IF CF=7.5 THEN LET CA=CF
 330 LET L=LEN D$
 340 LET P$="1"
 350 LET C$=""
 360 LET D$=C$
 370 LET BAN=1
 380 IF CF<>CA THEN GOTO 450
 390 GOSUB 1090
 400 LET COMP=COMP+B(I,1)
 410 PRINT AT 14,0;D$
 420 IF COMP>=CA THEN GOTO 760
 430 IF COMP>7.5 THEN GOTO 690
 440 GOTO 390
 450 IF CA>=7.5 THEN GOTO 760
 460 GOSUB 1090
 470 LET COMP=COMP+B(I,1)
```





Acerte a mira
O navio pode ser atingido em três regiões, e os pontos variam de 0 a 30. O resultado será bom se você conseguir 150 pontos ou mais.

```
1128 GOTO 1110
1130 INVERSE: HOME:VTAB 3:HTAB 11:PRINT
     "*** ATAQUE AEREO ***:NORMAL:VTAB 7
1140 PRINT "VOCE TEM 15 BOMBAS."
1145 PRINT: PRINT "A VELOCIDADE DO NAVIO
     VARIA, PORTANTO TOME CUIDADO PARA
     NAO DESPERDICAR BOMBAS."
1150 PRINT: PRINT: PRINT TAB (11)" B O A
     S O R T E"
1155 NORMAL
1160 VTAB 23
1165 INVERSE
1170 PRINT "TECLE <RETURN> PARA
     CONTINUAR"
1175 IF PEEK (-16384)=141 THEN POKE
     -16368,0:GOTO 1180
1178 GOTO 1160
1180 NORMAL:RETURN
2000 REM *** CRIAR VARIAVEIS ***
2020 TS=0
2030 SL=15
2050 GR:COLOR=2
2060 HLIN 0,39 AT 39
2080 HLIN 0,39 AT 38
2090 AX=0:SX=33:SS=1
2100 RETURN
3000 REM *** O JOGO ***
3030 HOME
3035 INVERSE
3040 VTAB 22:CALL -958:PRINT "TIROS
     FALTANDO";:NORMAL:PRINT" ";SL;" ";:
     INVERSE: PRINT "PONTOS":NORMAL
     PRINT " ";TS
3050 GOSUB 3090:REM AVIAO
3060 GOSUB 3130:IF SL=0 THEN RETURN
3070 GOSUB 3270:REM NAVIO
3080 GOTO 3050
3090 COLOR=0:HLIN AX,AX+6 AT 2:HLIN
     AX+1,AX+6 AT 1:PLOT AX+6,0
```



```
3100 REM
3110 AX=AX-1: IF AX<0 THEN AX=33
3120 COLOR=4:HLIN AX,AX+6 AT 2:HLIN
     AX+1,AX+6 AT 1:PLOT AX+6,0:RETURN
3130 IF FF THEN 3160
3140 IF PEEK (-16384)=<128 THEN RETURN
3150 FF=1:POKE -16368,0:FX=AX+3:FY=2
3160 COLOR=0:PLOT FX,FY
3170 FY=FY+1
3180 IF SCRN (FX,FY)=0 THEN
     COLOR=13:PLOT FX,FY:RETURN
3190 IF SCRN (FX,FY)=2 THEN 3230
3200 TS=TS+(38-FY)*10
3210 SC=0:COLOR=0:HLIN SX,SX+6 AT
     37:HLIN SX+3, SX+5 AT 36 : PLOT
     SX+4,35
3220 SX=33:SS=1:SC=0
3230 COLOR= 2:HLIN FX-1,FX+1 AT 37: PLOT
     FX-2,36:PLOT FX,36: PLOT FX+2,36 :
     PLOTFX-3,35:PLOT FX,35:PLOT FX+3,35
3240 COLOR= 0: HLIN FX-1,FX+1 AT 37:
     PLOT FX -2,36: PLOT FX,36: PLOT
     FX+2,36:PLOT FX-3,35:PLOT FX,35:
     PLOT FX+3,35
3250 FF=0:SL=SL-1:POKE -16368,0
3255 INVERSE
3260 VTAB 22: CALL -958:PRINT "TIROS
     FALTANDO";:NORMAL: PRINT " ";SL;"
     ";:INVERSE: PRINT "PONTOS";:NORMAL
     PRINT  " ";TS
3270 SC=SC+1:IF SC=SS THEN RETURN
3280 SC=0:COLOR=0:HLIN SX,SX+6 AT 37:
     HLIN SX+3,SX+5 AT 36:PLOT SX+4,35
3290 SX=SX+1: IF SX=> 33 THEN
     SX=0:SS=INT( RND(1)*3)+1:SC=SS
3300 COLOR=1:HLIN SX,SX+6 AT 37:HLIN
     SX+3,SX+5 AT 36: PLOT
     SX+4,35:RETURN
4000 REM *** FIM DE JOGO ***
4010 FOR N= 1 TO 3000:NEXT N
4020 TEXT: HOME: VTAB 3:HTAB 7: INVERSE:
     PRINT "*** ATAQUE AEREO ***"
4025 NORMAL
4030 PRINT:PRINT:PRINT "O JOGO
     ACABOU.":PRINT:PRINT:PRINT "VOCE
     OBTEVE ";:INVERSE:PRINT TS;:NORMAL:
     PRINT "PONTOS QUE E' ";
4040 IF TS =<25 THEN PRINT "HORRIVEL ":
     RETURN
4050 IF TS =<50 THEN PRINT "PESSIMO!!":
     RETURN
4060 IF TS =<75 THEN PRINT "RUIM!!! ":
     RETURN
4070 IF TS =<100 THEN PRINT "MEDIO! ":
     RETURN
4080 IF TS =<150 THEN PRINT "BOM!!! ":
     RETURN
4090 IF TS =<250 THEN PRINT "OTIMO!!!":
     RETURN
4100 IF TS =<450 THEN INVERSE: PRINT "":
     PRINT "FANTASTICO !!!":NORMAL:
     RETURN
```

INVASOR DO ESPAÇO

PARA LINHA SINCLAIR

O invasor extraterreno quer atingir seu canhão. Procure explodi-lo antes que ele chegue a Terra. Acertando-o quando ainda estiver bem alto, você ganhará mais pontos e evitará uma aproximação perigosa. O invasor pode tomar a Terra quatro vezes. A cada invasão, você perderá um canhão. Mas, fazendo 3.000 pontos, terá direito a bônus e mais um canhão. Observe os comandos: "8" movimenta o canhão para a direita; "5" move-o para a esquerda; com "O" você atira; e "C" reinicia o jogo. Para aumentar a dificuldade, basta mudar a linha 500 B = 3000 para um valor mais alto em B.

```
 10 REM **********************
 15 REM * INVASOR DO ESPACO *
 20 REM **********************
 30 GOTO 490
 40 FOR L=U TO 19
 50 LET X=X+(INKEY$="8" AND X(26) -(INK
    EY$="5" AND X)3)
 60 PRINT AT 19,X;" I$** "
 70 PRINT AT L-1,C;"    "
 80 LET C=C+A(L)
 90 PRINT AT L,C;" O "
 92 PRINT AT L,C;"    "
100 IF INKEY$="0" THEN GOSUB 160
110 NEXT L
120 LET F=F-1
130 LET Z=0
140 IF F(=0 THEN GOTO 895
150 GOTO 790
160 FOR M=17 TO L STEP -2
170 PRINT AT M,X+2;"."
180 PRINT AT M+2,X+2;" "
190 NEXT M
210 IF C=X+1 THEN GOTO 230
220 RETURN
230 FOR K=1 TO 20
240 FAST
250 SLOW
260 NEXT K
270 FOR K=1 TO 5
280 PRINT AT L,C-1;" BUM ";AT L,C-1;"
    "
290 NEXT K
300 LET Z=1
310 LET J=10*(INT (1/L*(L+10)*1000))
320 PRINT AT L,C-1;J
330 LET H=C
```

Emoção garantida
As opções de comando e a possibilidade de dificultar a conquista de pontos garantem momentos de verdadeira emoção.

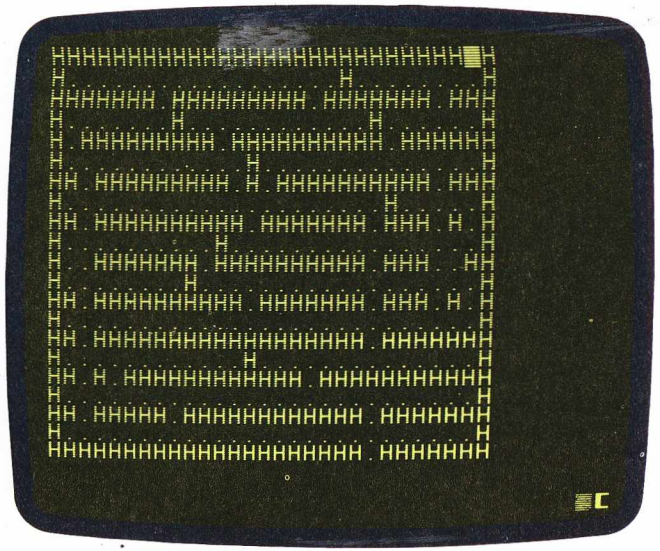
```
340 LET P=P+J
350 GOSUB 880
360 IF P>=B THEN GOTO 390
370 LET U=U+1
380 GOTO 790
390 LET J=500*F
400 PRINT AT L,C-1;"      ";AT 10,16;J
410 PRINT AT 10,10;"BONUS";
420 FOR N=1 TO 100
430 NEXT N
440 IF U>4 THEN LET U=U-4
450 LET B=B+3000
460 LET F=F+1
470 PRINT AT 10,10;"
480 GOTO 40
490 DIM A(25)
495 LET Z=0
500 LET B=3000
505 LET F=4
510 LET P=0
520 LET U=1
530 LET L=0
540 LET A=0
550 LET X=15
560 PRINT AT 3,3;"** INVASOR DO ESPACO
    **"
570 PRINT AT 10,1;"PRESS. QUALQUER
    TECLA PARA"
575 PRINT AT 12,10;"INICIAR"
640 IF INKEY$="" THEN GOTO 640
645 PRINT "QUE A FORCA ESTEJA COM VOCE"
650 FOR K=1 TO 20
660 SCROLL
670 PRINT AT 21,(RND*31);"*"
690 PRINT AT 21,(RND*31);"."
710 PRINT AT 21,(RND*31);"+"
720 NEXT K
730 SCROLL
740 SCROLL
750 SCROLL
770 SCROLL
790 GOSUB 880
800 IF Z=1 THEN PRINT AT L,H-1;"  "
810 FOR I=1 TO 20
820 LET A(I)=INT (RND*3)-1
830 LET X=X+(INKEY$="8" AND X(26) -(INK
    EY$="5" AND X)3)
840 PRINT AT 19,X;" I$** "
850 NEXT I
860 LET C=INT (RND*20)+6
870 GOTO 40
880 PRINT AT 21,2;P;" PONTOS";TAB
    21;F;" FOGUETES"
885 PRINT AT 20,0;"----------
    ----------"
890 RETURN
895 GOSUB 880
900 PRINT AT 10,10;" FIM DE JOGO "
905 PRINT AT 12,4;" NOVA PARTIDA TECLE
    (C)"
910 IF INKEY$()"C" THEN GOTO 910
920 CLS
930 RUN
```


ATAQUE AÉREO
PARA LINHA APPLE

Este jogo vai levá-lo ao oceano pilotando um avião de guerra. Sua missão é afundar todos os navios inimigos que atravessarem a rota do avião. Mas, cuidado! Os navios estarão cruzando o mar com velocidades variáveis e, além disso, atingi-los na proa ou na popa não vai afundá-los, apenas danificá-los. Por isso, calcule bem sua velocidade e a trajetória dos projéteis, a fim de acertá-los no centro, onde o número de pontos que você ganha é maior.
Depois de dez navios atingidos, você obterá sua contagem de pontos. Procure superar a marca anterior.

Navios velozes
A grande dificuldade deste jogo reside na velocidade variável dos navios, o que provoca, muitas vezes, um desperdício de bombas.

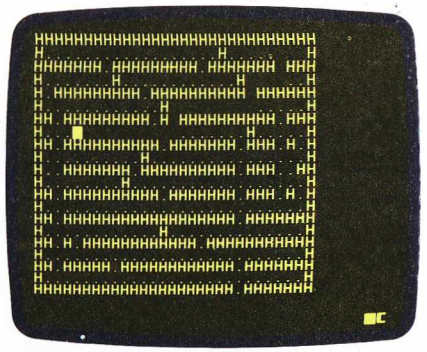
```
10    REM ATAQUE AEREO
80    GOSUB 1000:REM INSTRUCOES
90    GOSUB 2000:REM CRIA VARIAVEIS
100   GOSUB 3000:REM INICIO DE JOGO
110   GOSUB 4000:REM FIM DE JOGO
120   VTAB 21:HTAB 1:INVERSE
130   PRINT "VOCE QUER JOGAR NOVAMENTE ?
      (S/N) ";GET A$:NORMAL
140   IF A$="S" THEN GOTO 90
150   IF A$<> "N" THEN  GOTO 120
160   VTAB 21:HTAB 1: PRINT SPC(39):END
1000  REM *** INSTRUCOES ***
1030  TEXT:HOME
1035  INVERSE
1040  VTAB 3:HTAB 11:PIRNT "*** ATAQUE
      AEREO ***"
1045  NORMAL
1050  PRINT: PRINT: PRINT "VOCE E UM
      PILOTO DE COMBATE."
1060  PRINT: PRINT "PONTOS SAO OBTIDOS
      QUANDO UMA DAS SUAS BOMBAS, ACERTA
      O NAVIO, QUE TEM QUE AFUNDAR."
1070  PRINT
1080  PRINT "PARA SOLTAR UMA BOMBA APERTE
      QUALQUER TECLA. O NUMERO DE PONTOS
      OBTIDOS DEPENDERA DA PARTE DO NAVIO
      QUE FOI ATINGIDO."
1090  PRINT
1100  PRINT "A PARTE INFERIOR DO NAVIO =>
      10 PONTOS"
1105  PRINT "A PARTE SUPERIOR DO NAVIO =>
      20 PONTOS"
1108  PRINT "A CHAMINE DO NAVIO => 30
      PONTOS"
1110  VTAB 23
1120  PRINT "TECLE <RETURN> PARA
      CONTINUAR"
1125  IF PEEK (-16384)=141 THEN POKE
      -16368,0: GOTO 1130
```



Começa a perseguição
Representado pelo cursor, você está prestes a iniciar sua corrida pelo labirinto. Cuidado com o monstro!

Um inimigo implacável
Para vencer o monstro, é preciso muita calma, além da rapidez. Neste caso, por exemplo, você ainda precisa comer todos os pontos do canto superior à direita.

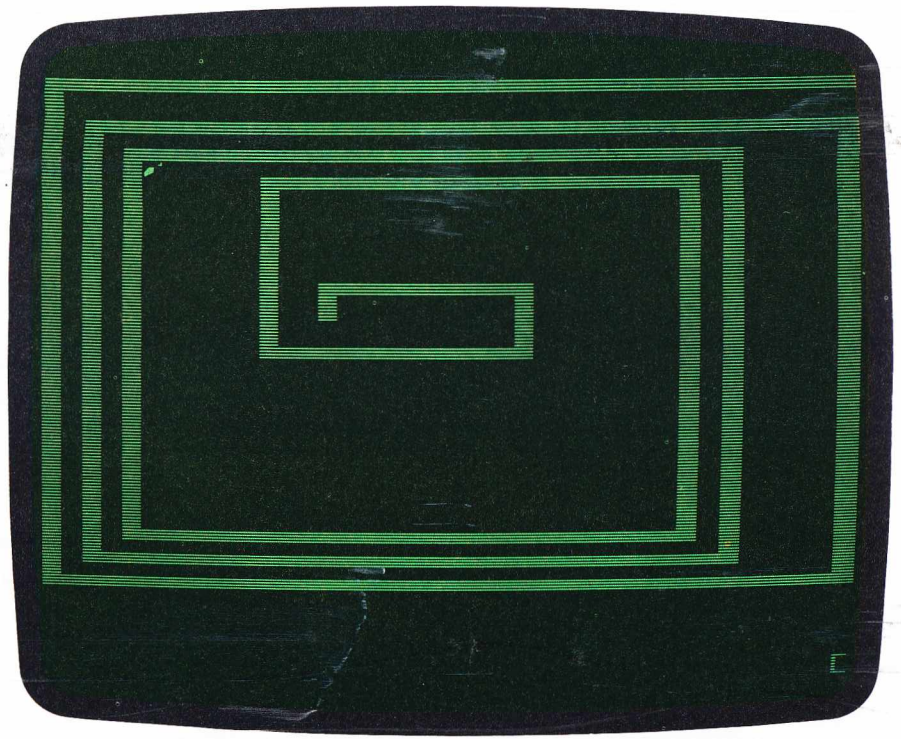
```
180 PRINT AT 7,15;".";AT 7,17;;"."
190 PRINT AT 15,15;".";AT 15,17;"."
200 PRINT AT 9,16;".";AT 13,16;"."
210 PRINT AT 11,3;".";AT 11,29;"."
220 PRINT AT 6,16;"H";AT 7,16;"H"
230 PRINT AT 11,16;"H";AT 15,16;"H"
240 PRINT AT 16,16;"H"
250 FOR N=2 TO 13 STEP 2
260 PRINT AT N,4;"."
270 PRINT AT N,28;"."
280 NEXT N
290 FOR N=8 TO 14 STEP 2
300 PRINT AT N,3;"H"
310 PRINT AT (N+1),29;"H"
320 NEXT N
330 PRINT AT 0,0;
340 LET C=PEEK 16398+256*PEEK 16399+115
350 LET P=C+495
360 LET S=27
370 SLOW
390 LET B=6.5
400 LET X=0
410 LET M$=INKEY$
420 IF M$()"" THEN LET B=VAL M$
430 POKE C,S
440 POKE P,0
450 LET X=34*B-50*SGN (B-6.5)-221
460 IF PEEK (P+X)()45 THEN LET P=P+X
470 POKE P,128
480 LET Y=33*SGN (INT ((P-C)/28))
490 IF PEEK (C+Y)()45 THEN LET C=C+Y
500 LET Y=SGN ((ABS ((P-C)/33-INT ((P-C
    )/33))(.5)-.5)
510 IF PEEK (C+Y)()45 THEN LET C=C+Y
520 LET S=PEEK C
530 POKE C,151
540 IF P=C THEN GOTO 560
550 GOTO 380
570 CLS
580 PRINT AT 0,0;"TENTE DE NOVO, PORQUE
    DESTA VEZ"
600 FOR N=1 TO 10
620 PRINT TAB 12;"NAO DEU"
630 NEXT N
640 PRINT "QUER TENTAR DE NOVO?
```



```
650 INPUT B$
660 IF B$="S" THEN GOTO 1
670 PRINT
680 PRINT
690 PRINT "TCHAU"
```


CRAM
PARA LINHA TRS-80

Neste jogo, o programa começa traçando uma linha que margeia a tela. Pouco antes de fechar a linha e completar o retângulo, você deve desviá-la (sempre para a direita), de modo a formar novo retângulo, interno ao primeiro. Procure fazer o maior número de retângulos e veja quantas vezes é capaz de desviar em tempo. O BASIC da linha TRS-80 Modelo III e seus compatíveis não incluem comandos de som; assim, há uma pequena rotina em linguagem de máquina responsável pela produção de efeitos sonoros. Caso seu microcomputador não possua alto-falante, você pode obter os efeitos ligando o cabo de saída do gravador a um amplificador.



Funções de trechos do programa

100-130: Inicializa a rotina de som, variáveis em geral, apresenta títulos e instruções, e atribui valores às variáveis principais — C (contador de curvas), I e J (coordenadas do ponto iluminado na tela).

180-580: Corpo principal do programa. Executa os quatro movimentos (direita, baixo, esquerda e cima) e verifica se a linha encontrou algum ponto iluminado (POINT I,J) e se alguma tecla foi usada (INKEY$).

Neste último caso, o programa passa ao próximo tipo de movimento e atualiza o contador de curvas. Se a linha encontrou um ponto iluminado passa-se ao fim do programa.

590-680: Desenha as quatro primeiras linhas, fazendo o contorno da tela.

690-780: Fim do jogo. Mostra quantas curvas foram feitas e toca a música de acordo com o número de curvas. Volta ao começo do loop principal para uma nova partida.

790-940: Apresenta os títulos e verifica se o usuário quer ver as instruções.

950-1080: Escreve as instruções na tela.

1090-1350: Inicializa rotina de som. Lê os dados das instruções DATA e faz o POKE (a inclusão) dos dados na área da memória reservada à variável A$. Note que, após rodar o programa, o valor atribuído a A$ será alterado. Note também que a versão cassete tem a linha 1220 alterada.

1280-1580: Rotinas de produção de sons. Você pode usar a rotina de inicialização e estas chamadas em outros programas de sua autoria.

```
10   REM ********************************
20   REM *                              *
30   REM *   CRAM PARA LINHA TRS-80     *
40   REM *                              *
50   REM ********************************
60   REM
70   REM ********************************
80   REM *        INICIALIZACAO         *
90   REM ********************************
100  GOSUB 1100
110  DEFINT A - Z
120  GOSUB 810
130  C=0:I=0:J=2
140  REM ********************************
150  REM *        COMECO DO JOGO        *
160  REM ********************************
170  GOSUB 620
180  REM ********************************
190  REM *           LOOP               *
200  REM ********************************
210  REM
220  REM ********************************
230  REM *    MOVIMENTO PARA DIREITA    *
240  REM ********************************
250  I=I+1
260  IF POINT(I,J) THEN GOTO 720
270  SET(I,J)
280  IF INKEY$="" THEN 250
290  C=C+1
300  X=USR(8305)
310  REM ********************************
320  REM *     MOVIMENTO PARA BAIXO     *
330  REM ********************************
340  J=J+1
350  IF POINT(I,J) THEN GOTO 720
360  SET(I,J)
370  IF INKEY$="" THEN GOTO 340
380  C=C+1
390  X=USR(8319)
400  REM ********************************
410  REM *   MOVIMENTO PARA ESQUERDA    *
420  REM ********************************
430  I=I-1
440  IF POINT(I,J) THEN GOTO 720
450  SET(I,J)
460  IF INKEY$="" THEN GOTO 430
470  C=C+1
480  X=USR(8343)
490  REM ********************************
500  REM *     MOVIMENTO PARA CIMA      *
510  REM ********************************
520  J=J-1
530  IF POINT(I,J) THEN GOTO 720
540  SET(I,J)
550  IF INKEY$="" THEN GOTO 520
560  C=C+1
570  X=USR(8335)
580  GOTO 250
590  REM ********************************
600  REM *         DESENHA TELA         *
610  REM ********************************
620  CLS
630  FOR A=0 TO 127:SET(A,0):NEXT A
640  FOR A=0 TO 47:SET(127,A):NEXT A
650  FOR A=127 TO 0 STEP -1:SET(A,47):
       NEXT A
660  FOR A=47 TO 2 STEP -1:SET(0,A):
      NEXT A
670  A$=INKEY$
680  RETURN
690  REM ********************************
700  REM *          FIM DO JOGO         *
710  REM ********************************
720  GOSUB 1380 :IF C<10 THEN B$="! "
         ELSE B$="!"
730  PRINT @476, B$ ; C ; " !";
740  IF C=1 THEN C$="VOLTA" ELSE C$=
    "VOLTAS"
750  PRINT @540, "!"; C$ ; "!";
760  IF C>20 THEN GOSUB 1480
770  IF C>30 THEN GOSUB 1480 :GOSUB 1480
780  FOR A=1 TO 1500:NEXT:GOTO 130
790  REM ********************************
800  REM *           TITULOS            *
810  REM ********************************
820  CLS
830  FOR A=1 TO 15
840  PRINT TAB(6+A*2) "C R A M"
850  NEXT A
860  GOSUB 1380 :GOSUB 1380 :PRINT
       CHR$(23);
870  FOR B=1 TO 1500:NEXT B
880  CLS
890  PRINT TAB(8) "BENVINDO AO JOGO ";
      "< CRAM > !"
900  PRINT TAB(8) "LIGUE O SEU MICRO ";
       "A UM AMPLIFICADOR"
910  PRINT:PRINT "VOCE QUER INSTRUCOES?"
920  A$=INKEY$:IF A$ = "" THEN GOTO 920
930  IF A$="N" THEN GOTO 1070
940  IF A$<>"S" THEN GOTO 920
950  REM ********************************
960  REM *          INSTRUCOES          *
970  REM ********************************
980  PRINT:PRINT  "O OBJETIVO DO "
      ;"JOGO E VIRAR O MAIOR NUMERO"
      ;"DE VEZES POSSIVEL NA   TELA.";
990  PRINT  " CADA VEZ QUE VOCE ";
     "QUISER MUDAR A DIRECAO DA LINHA";
    " QUE SE  MOVE. "
1010 PRINT  "USE QUALQUER TECLA";
      " EXCETO <SHIFT> OU <BREAK>."
1020 PRINT  "O JOGO TERMINA QUANDO ";
      "A LINHA BATER EM ALGUMA"
1030 PRINT  "LINHA JA' DESENHADA."
      :PRINT"AO FINAL DO JOGO VOCE "
1040 PRINT  "VERA' O NUMERO DE";
      :PRINT" VEZES QUE CONSEGUIU DESVIAR.
1050 PRINT:PRINT  "TECLE <ENTER> ";
      :PRINT "PARA COMECAR ..."
1060 A$=INKEY$:IF A$="" THEN GOTO 1060
1070 CLS
1080 RETURN
1090 REM ********************************
1100 REM *  INICIALIZA ROTINA DE SOM    *
1110 REM *      (AA$ = 22 ESPACOS)      *
1120 REM ********************************
1130 AA$="                      "
1140 QA!=VARPTR (AA$)
1150 QB!=PEEK(QA!+1)+PEEK(QA!+2)*256
1160 REM ********************************
1170 REM *     PARA  VERSAO CASSETTE     *
1180 REM *     MUDAR LINHA 1220 PARA     *
1190 REM *     POKE 16526,PEEK(QA+1):    *
1200 REM *     POKE 16527,PEEK(QA+2)     *
1210 REM ********************************
1220 DEFUSR=QB!
1230 FOR A!=QB! TO QB!+21
1240 B!=A!:IF B!>32767 THEN B!=B!-65536
1250 READ Q:POKE B!,Q
1260 NEXT A
1270 RETURN
1280 REM ********************************
1290 REM *      LINGUAGEM DE MAQUINA     *
1300 REM *      PARA ROTINA DE SOM       *
1310 REM ********************************
1320 DATA 205,127,10,76,69,62,1
1330 DATA 211,255,16,254,69,62,16
1340 DATA 211,255,16,254,13,32,239,201
1350 REM ********************************
1360 REM *       SOM DO FIM DO JOGO      *
1370 REM ********************************
1380 FOR A=1 TO 3
1390 X = USR(8319)
1400 X = USR(8343)
1410 X = USR(8335)
1420 X = USR(8305)
1430 NEXT A
1440 RETURN
1450 REM ********************************
1460 REM *     SOM MAIS DE 20 VOLTAS     *
1470 REM ********************************
1480 X = USR(32383)
1490 X = USR(32383)
1500 X = USR(14479)
1510 X = USR(13719)
1520 X = USR(27287)
1530 X = USR(12966)
1540 X = USR(13719)
1550 X = USR(27287)
1560 X = USR(27287)
1570 X = USR(27287)
1580 RETURN
```

COME-COME
PARA A LINHA SINCLAIR

Você tem de percorrer o labirinto, comendo os pontos à medida em que avança. Em seu encalço, vem um monstro ameaçador. Você vence o jogo se conseguir percorrer todo o labirinto e apagar todos os pontos sem que o monstro o devore. Se o monstro for mais rápido, você perde a rodada. O programa vai construir, com as letras "H", as paredes do labirinto, e colocar nos corredores uma linha de pontos. Em seguida, alguns "H" e pontinhos são intercambiados para formar os caminhos internos do labirinto. Aparece então um cursor no canto superior esquerdo da tela. Este percorre o labirinto conforme você use estas teclas: 5 (que o movimenta para a esquerda), 6 (para baixo), 7 (para cima) e 8 (para a direita).

```
 1 REM ************************
 2 REM * PARA A LINHA SINCLAIR *
 3 REM *      COME - COME      *
 4 REM ************************
10FAST
20 PRINT
30 FOR I=1 TO 10
40 PRINT "HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
   HHH"
50 PRINT "H.........................
   ..H"
60 NEXT I
70 PRINT "HHHHHHHHHHHHHHHHHHHHHHHHHHHHH
   HHH"
80 FOR N=1 TO 30 STEP 14
90 FOR J=3 TO 19 STEP 4
100 PRINT AT J,N;"."
110 NEXT J
120 NEXT N
130 FOR N=5 TO 17 STEP 6
140 PRINT AT N,11;".";AT N,21;"."
150 NEXT N
160 PRINT AT 9,29;".";AT 9,29;"."
170 PRINT AT 13,2;".";AT 13,29;"."
```